# Data-driven Crowd Modeling Techniques: A Survey

JINGHUI ZHONG, South China University of Technology, China and China-Singapore International Joint Research Institute, China
DONGRUI LI, ZHIXING HUANG, and CHENGYU LU, South China University of Technology, China
WENTONG CAI, Nanyang Technological University, Singapore

Data-driven crowd modeling has now become a popular and effective approach for generating realistic crowd simulation and has been applied to a range of applications, such as anomaly detection and game design. In the past decades, a number of data-driven crowd modeling techniques have been proposed, providing many options for people to generate virtual crowd simulation. This article provides a comprehensive survey of these state-of-the-art data-driven modeling techniques. We first describe the commonly used datasets for crowd modeling. Then, we categorize and discuss the state-of-the-art data-driven crowd modeling methods. After that, data-driven crowd model validation techniques are discussed. Finally, six promising future research topics of data-driven crowd modeling are discussed.

CCS Concepts: • **General and reference** → *Surveys and overviews*; • **Computing methodologies** → **Model development and analysis**

Additional Key Words and Phrases: Crowd simulation, crowd model validation, agent-based crowd modeling, data-driven crowd modeling

**ACM Reference format:**
Jinghui Zhong, Dongrui Li, Zhixing Huang, Chengyu Lu, and Wentong Cai. 2022. Data-driven Crowd Modeling Techniques: A Survey. *ACM Trans. Model. Comput. Simul.* 32, 1, Article 4 (January 2022), 33 pages.
https://doi.org/10.1145/3481299

4

## 1 INTRODUCTION

Crowd modeling is an active research field that has attracted increasing attention from both academia and governments. It has numerous practical applications, including, but not limited to, abnormal behavior detection [1, 2], military training [3, 4], and game design [5–8]. Generally, crowd modeling aims to define the crowd behaviors in given scenes, and the defined models may

This work is supported by the National Natural Science Foundation of China (Grant No. 62076098), the Guangdong Natural Science Foundation Research Team (Grant No. 2018B030312003), and the Fundamental Research Funds for the Central Universities (Grant No. D2191200).
Authors' addresses: J. Zhong, South China University of Technology, Guangzhou, China, China-Singapore International Joint Research Institute, Guangzhou, China; email: jinghuizhong@scut.edu.cn; D. Li, Z. Huang, and C. Lu, School of Computer Science and Engineering, South China University of Technology, China; emails: li_dongrui@foxmail.com, zhixing.huang@ecs.vuw.ac.nz, chengyulu3-c@my.cityu.edu.hk; W. Cai, School of Computer Science and Engineering, Nanyang Technological University, Singapore; email: aswtcai@ntu.edu.sg.

1049-3301/2022/01-ART4 $15.00
https://doi.org/10.1145/3481299

be applied in crowd simulation [9]. Compared with other types of modeling techniques, crowd modeling is relatively challenging, because crowd behavior is influenced by a number of factors, such as physical, psychological, and social factors [9]. Moreover, crowd modeling has its own distinct characteristics, because it involves components of model steering behavior, path planning behavior, and collision avoidance behavior of pedestrians.

Crowd modeling methods include macroscopic and microscopic methods. Macroscopic methods treat the crowd as a whole and pay little attention to individuals, such as flow-based methods. Microscopic methods focus on individual behaviors, including particle-based methods, cellular automation methods and agent-based methods. Compared to other methods, agent-based crowd modeling methods take more social and psychological factors into consideration to generate crowds with better realism.

How to define realistic crowd behavior is a fundamental problem in crowd modeling. Traditional trial-and-error modeling methods are time-consuming and tedious and require much manual effort. In recent years, with the rapid development of **closed-circuit television (CCTV)** systems and video processing techniques, data-driven crowd modeling, which constructs realistic crowd models by using crowd data [10], has become more prevalent. In this survey, we focus on data-driven crowd modeling techniques that utilize real data to assist model construction and model validation.

To date, various data-driven crowd modeling techniques have been proposed, and these methods differ from each other in terms of factors such as the data utilized, the methods used for analyzing data and the patterns learned from data. A number of crowd datasets, with each having different formats and features, are available in the literature for learning crowd behaviors, such as the BIWI [11] dataset and the avoidance behavior dataset [12]. With real crowd data, data-driven modeling methods are capable of generating more realistic crowd behavior compared to traditional manual methods [5, 13–17]. Some data-driven modeling methods focus on tuning the parameters of crowd models [15, 16], while other methods focus on extracting regular patterns from data and using the extracted patterns to generate new crowd simulation [5, 17]. To facilitate automatic crowd modeling, multiple validation methods have also been proposed to measure the quality of crowd models [18–20].

Although numerous techniques related to data-driven crowd modeling have been proposed over the past decades, there is still a lack of a comprehensive survey on this research direction. There are several related surveys in the literature, but these surveys only focus on certain aspects of crowd modeling. For example, Li et al. [21] focused on surveying techniques for analyzing crowd data, such as pattern segmentation and behavior recognition, but they did not mention methods for applying the learned features to construct crowd models. Grant et al. [22] focused on crowd video analysis and video datasets and paid less attention to crowd model construction. Dong et al. [23] surveyed pedestrian and evacuation dynamics from four aspects: data collection, characteristic extraction, modeling and validation, and application. However, they only briefly mentioned several classic data-driven modeling methods. Unlike existing surveys, this article aims to provide a comprehensive survey on both state-of-the-art and classic data-driven crowd modeling techniques, from multiple perspectives, including the datasets, model construction methodologies, model validation methods, and software systems. The purpose of this article is to facilitate researchers improving their crowd models by using crowd data.

As the term "data-driven" indicates, data constitute the cornerstone of data-driven modeling methods. The term "data" here means crowd data from the real world. In the literature, most crowd datasets contain raw visual material, such as videos [24–26]. For these types of crowd datasets, researchers often need to extract motion features or trajectories before applying them to model construction. Meanwhile, some crowd datasets are provided along with processed data, such as trajectory records and annotations [27, 28]. These types of crowd datasets are more convenient

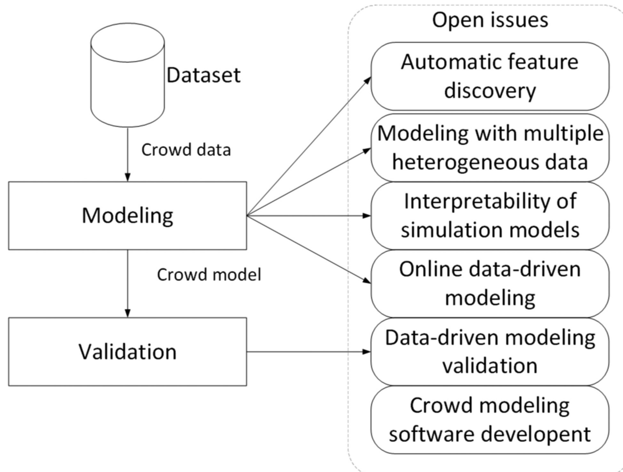


Fig. 1. Relationships among data, modeling, validation, and open issues.

for researchers to use than the raw ones. In this article, we identify dozens of well-known crowd datasets and discuss the features of the crowd datasets.

As for the data-driven modeling methodologies, we are interested in what knowledge is learned from data and how they learn that knowledge from data. Thus, we discuss the data-driven crowd modeling methods in two dimensions. We first discuss what can be learned from data, such as parameter configurations [15, 29], state-action pairs [14, 30], and behavior rules [31–33]. Then, we discuss their learning approaches, such as learning with mathematical models [16, 34] and machine learning techniques [5, 35].

How to validate crowd models based on crowd data and crowd simulation results is of fundamental importance for data-driven modeling. In this article, we categorize the existing crowd validation methods into three types. The first is metric-based validation methods, which use a set of metrics to compare the simulated crowd and the real data [36–38]. The second is probability-based validation methods, which establish a probability model to validate the algorithm performance [19, 39]. The other validation methods [40, 41] are designed flexibly by researchers to investigate certain aspects of crowd models.

Despite the large amount of work done by the research community, as this survey discusses in the following sections, there are still open research issues in data-driven crowd modeling. In this article, we present six open issues, including automatic feature discovery, modeling with heterogeneous data, model interpretability improvement, online modeling and effective validation. In this article, we also discuss the development of crowd modeling software and suggest possible ways to incorporate the "data-driven" pattern into the industrial software.

Figure 1 illustrates the relationships among datasets, crowd modeling, model validation, and open issues discussed in this article. The first step of data-driven crowd modeling is to collect datasets for model training and validation. After the model is trained, simulation results of the models are utilized to validate the model. The open issues are related to both crowd modeling and validation. Specifically, the following open issues regarding crowd modeling are discussed: data acquisition for model development, fusion of data from multiple sources in modeling process, interpretability of learnt simulation models, and utilization of online learning for model adaptation. Finally, to provide a complete picture, open issues in crowd modeling software development are discussed.

The rest of this article is organized as follows: Section 2 introduces typical crowd datasets for data-driven crowd modeling. Then, Section 3 presents the data-driven crowd modeling methodologies in terms of what knowledge they learn from data and how they learn it. After that, Section 4 discusses the data-driven crowd validation methods, followed by a discussion of open research issues in Section 5. Finally, Section 6 draws a conclusion.

## 2 DATASETS FOR CROWD MODELING

In the past decades, a number of crowd datasets have been released. However, the attributes of different datasets still lack a clear summary. In this section, we discuss different features of crowd datasets, including the format, applications, and availability of the datasets, to help the community have a more comprehensive understanding. We collect the datasets from the existing literature and websites of different research groups. These surveyed datasets are listed in Table 1. For brevity, we consider three categories of data formats: raw video data, tracking data, and annotations.

Most datasets provide data in raw video format. This type of data provides top-down views, helping researchers to identify trajectories of pedestrians, or low-angle views, describing detailed pedestrian behaviors such as turning their head. There are several typical crowd datasets of raw video data, such as INRIA [45], UCSD [42], BIWI [11], and WorldExpo'10 [49]. INRIA [45] is a video dataset of a dense crowd that contains videos in both the top-down view and the low-angle view. It has been popularly used for tracking and event detection. UCSD [42] is a dataset of gray pictures captured in a low-angle view, which has been used for video clustering and segmenting. BIWI [11] contains a video captured in a top-down view, which has been used as a dataset for tracking. WorldExpo'10 [49] is a large and high-resolution video dataset for region-level crowd segmentation and collectiveness analysis. These video datasets can also be re-annotated by researchers for specific purposes.

The second type of datasets is tracking data, which records the trajectories of pedestrians extracted from videos. For example, CBE [43], a dataset that is widely used for tracking and path prediction, contains pedestrian trajectories recorded by splines represented by a series of control points. CVBASE and the train station [46] datasets include both videos and trajectories. The avoidance behavior dataset [12] records the tracks of pedestrians who avoid obstacles or other pedestrians. This dataset provides collision avoidance patterns of pedestrians. The diluted pedestrian dynamics dataset contains a great number of trajectories, aiming at enabling ensemble analysis of diluted pedestrian motion. The collective motion database [48] contains images and track files, which are used to study human collective behaviors. The tracking data, compared with raw video data, are more convenient for researchers to investigate the movement of pedestrians.

Annotation datasets are another type of data for crowd modeling, aiming to help researchers investigate specific aspects of pedestrian behaviors. In the BIWI dataset [11], the pedestrian positions are annotated, and the grouping behaviors are labeled. The Stanford drone dataset [50] provides person annotations, as well as their categories. The UMD dataset [2] contains different event labels and abnormal crowd behaviors. In addition to the datasets mentioned above, there are datasets that provide statistical information for special applications, such as the gender and age distributions of crowds in SGVDS [56].

## 3 DATA-DRIVEN CROWD MODELING METHODOLOGIES

Before modeling, crowd analysis is necessary for understanding crowd data, which influences how datasets are used. Crowd analysis is the procedure of describing training crowd data by specified features [59]. Crowd features include macro features and micro features. Macro features describe the crowd as a whole or several parts, regardless of individual personality, while micro features focus on individual moving patterns in given data. There are many approaches to calculate these features, and Li et al. [21] have discussed them. Because our major theme is data-driven crowd

Table 1. Common Datasets for Crowd Modeling

| Name | Data format | Application | Website or URL |
|---|---|---|---|
| UMD [2] | video, event annotation | event prediction | http://mha.cs.umn.edu/proj_events.shtml#thrown, and Getty Images, ThoughtEquity.com |
| DSBM [24] | video, event annotation | event detection | The data sets are available for public use upon request to authors |
| BIWI [11] | video, annotation | tracking | http://www.vision.ee.ethz.ch/en/datasets/ |
| UCSD [42] | frame | video cluster, segmenting | http://www.svcl.ucsd.edu/projects/motiondytex/ |
| CBE [43] | video, track | tracking, prediction, sparse,dense | https://graphics.cs.ucy.ac.cy/research/downloads/crowd-data |
| UCF tracking in high density crowds dataset [44] | frame | dense | https://www.crcv.ucf.edu/data/tracking.php |
| INRIA [45] | video | tracking, event detection | http://www.mikelrodriguez.com/datasets-and-source-code/#datadriven |
| Shopping mall [26] | frames, ground truth, normalized features, perspective map | tracking, counting, crowd counting, profiling research | http://personal.ie.cuhk.edu.hk/~ccloy/downloads_mall_dataset.html |
| Train station [46] | video, tracked | tracking | http://mmlab.ie.cuhk.edu.hk/project/dynamicagent/ |
| UCF crowd segmentation dataset [47] | video | | https://www.crcv.ucf.edu/data/crowd.php |
| Collective Motion Database [48] | frames, track | crowd collectiveness | http://mmlab.ie.cuhk.edu.hk/projects/collectiveness/dataset.htm |
| WorldExpo'10 [49] | video, annotation | segmentation, density, collectiveness, cohesiveness | http://www.ee.cuhk.edu.hk/~xgwang/crowdexpo.html |
| Stanford Drone Dataset [50] | video, annotation | tracking, trajectory prediction | http://cvgl.stanford.edu/projects/uav_data/ |
| i-LIDS [51] | footage | event detection, tracking scenarios | https://www.gov.uk/guidance/imagery-library-for-intelligent-detection-systems#i-lids-datasets |
| Gallery Dataset [52] | density, trajectories, presence of groups, spatial arrangement, walking speed | group/density analysis, trajectory prediction | http://www.csai.disco.unimib.it/CSAI/GalleryDataset/(need additional E-mail) |
| PETS [25] | frame, ground truth | counting, tracking, event prediction, density estimation | http://www.cvg.reading.ac.uk/PETS2009/a.html |
| ETISEO [53] | various | video processing | http://www-sop.inria.fr/orion/ETISEO/download.htm#video_data |
| Time Square Intersection [54] | frames, weather data, traffic data | video classification, tag annotation | http://www.eecs.qmul.ac.uk/~xiatian/downloads_qmul_TISI_dataset.html |
| Educational Resource Centre (ERCe) [54] | frame, campus event schedule | video classification, annotation,clustering | http://www.eecs.qmul.ac.uk/~xiatian/downloads_qmul_ERCe_dataset.html |
| OpenPTDS [27] | video, track | evacuation | http://shi.buaa.edu.cn/songxiao/en/lwcg/28058/content/7381.htm#lwcg |
| CAVIAR | frames, annotated | event prediction, boxing | http://homepages.inf.ed.ac.uk/rbf/CAVIAR/ |
| BEHAVE | video, annotated | event detection | http://groups.inf.ed.ac.uk/vision/BEHAVEDATA/INTERACTIONS/ |
| CVBASE | video, tracked, annotated | event detection | http://vision.fe.uni-lj.si/cvbase06/downloads.html |
| Evacuation through a narrow door | video, tracked, time, spatio-temporal diagram | evacuation | http://ped.fz-juelich.de/extda/garcimartin2013 |
| Avoidance behavior [12] | annotation | collision avoidance, tracking | http://ped.fz-juelich.de/extda/parisi2016 |
| diluted pedestrian dynamics | tracked | | https://doi.org/10.4121/uuid:25289586-4fda-4931-8904-d63efe4aa0b8 |
| Edinburgh Informatics Forum Pedestrian Database [55] | histogram, track | tracking, targeting | http://homepages.inf.ed.ac.uk/rbf/FORUMTRACKING/ |
| SGVDS [56] [57] [58] | gender, age distribution, travel speeds | evacuation | https://fseg.gre.ac.uk/validation/ship_evacuation/ |
| "Central" Pedestrian Crossing Sequences [28] | frame, pedestrian annotation | tracking | http://www.vision.ee.ethz.ch/datasets_extra/iccv07-data.tar.gz |

modeling, discussion concerning crowd analysis, which is in an earlier stage than crowd modeling, is outside the scope of this article.

Traditional crowd modeling methods mainly rely on the knowledge of modelers. The main duty of modelers is to design an initial model structure in a manner that they consider reasonable and then tune the model using trial-and-error. Zhou et al. [60] summarized the common crowd modeling and simulation technologies, which are categorized according to crowd sizes and time-scales. In the last decades, to generate more realistic crowd behaviors, data-driven crowd modeling methods have been proposed, which differ from the traditional ones, because they learn knowledge (e.g., behavior patterns) from real data. In this section, we classify these data-driven modeling methods according to what they learn from real data and discuss how they work. The typical methods we discuss are listed in Table 2.

Crowd modeling and simulation have been a research topic for decades and related works have been proposed. Researchers and experts designed the forms and parameters of those models, which took a great effort. Besides, whether the proposed models are precise is a question, since they are created subjectively. Data-driven methods can release humans from heavy modeling work by constructing objective models based on real data. Generally, existing data-driven crowd modeling methods can be classified into three major types, as shown in Table 2. The first type of data-driven crowd modeling methods use the data to calibrate the parameters of existing models. They still adopt traditional models, but additionally, data are used to tune their parameters [61–63] to make the simulation more realistic. The second type of data-driven crowd modeling methods derive patterns from the data and use the derived patterns in simulation (i.e., mimic real situation). They adopt different types of modeling techniques, which require data during simulation as well as modeling. The combination of data-driven and general modeling methods reduce the burden of modelers, since crowd models need not to be designed from the beginning. Different from the second type of methods, which memorizes real data, the third type of data-driven crowd modeling methods learn knowledge from data (e.g., derive rules and use the derived rules in simulation). During simulation, agents can obtain their next movement according to their current state and the learned knowledge. This type of methods can mine behavior rules from data, which further lightens the work of modelers.

### 3.1 Calibration Methods

Data-driven calibration methods usually have model templates specified by modelers in advance. However, unlike traditional modeling methods, these methods learn unknown parameters related to the environment or behaviors of their predefined models from real data.

In some data-driven calibration methods, data only play a small role in the calibration procedure. The major part of the calibration work depends heavily on experts' domain knowledge. For example, the method proposed in Reference [15] learns parameters from real-life videos of three different scenarios. After determining the parameters to be optimized, each of these parameters is simply checked from its minimum value to its maximum value while other parameters are fixed. Then, each parameter is assigned the value that causes the smallest error between the calculated results and video data. Another method [29] analyzes the habits of human walk and the dependence of the next movement on the previous one from real experimental motion-captured data. It estimates the bounds of pedestrian velocity and acceleration after data analysis and obtains an admissible motion set for the next step of each pedestrian. A cost function is used to optimize the velocity change before the next movement. Brogan et al. [64] designed experiments to obtain pedestrian data, including desired speeds and heading charts. Then pedestrians' behaviors can be determined according to their states and the environment. Reference [65] proposed a cellular automaton model for crowd movement during evacuation, whose parameters are calibrated

Table 2. Methods for Data-driven Crowd Modeling

| Category | | Method | Reference | Description |
|---|---|---|---|---|
| Calibration | basic tuning | data matching | [15] | tune specific parameters to minimize the error between calculated results and real data |
| | | bounds determination | [29] | analyze data to generate bounds of parameters of pedestrian behaviors |
| | | feature learning | [64] [65] [66] [67] [68] | adjust parameters of existing models like the social force model, RVO2 and Gaussian distribution |
| | mathematical technique | gradient-based optimization | [16] | adjust parameters of optimization function to generate pedestrian behaviors similar to data |
| | | the least-squared method | [34] | recalibrate the model applying the least-squares method on participants¢ speed |
| | | maximum likelihood estimation | [69] | calibrate model parameters of interactions among pedestrians crossing each other according to experimental data |
| | | simplex algorithm | [70] | use a variant of simplex algorithm to optimize parameters of the energy function that encodes the model factors |
| | | log marginal likelihood | [71] | learn hyperparameters of Gaussian process that model pedestrian interactions from data |
| | evolutionary algorithm | general evolutionary methods | [72] [73] [74] | encode the parameters of specified models, evolve them, and finally decode the individuals to get parameters that can help models generate realistic crowd |
| | | genetic algorithm | [11] [75] [76] [77] | |
| | | differential evolution | [78] [18] [79] | |
| Example-based | database | state matching | [43] [80] [81] | find similar states in the database to get corresponding actions |
| | | fuzzy logic | [30] | add uncertainty to actions of similar states to improve realism |
| | | selected examples | [14] | only store important examples to reduce the database size |
| | mathematical technique | velocity field | [82] | transfer examples to fields that can be used to generate the animation |
| | | probability | [13] | generate probabilities of behaviors in different state |
| | | weighted linear regression | [81] | solve the problem of the lack of examples during matching caused by discrete examples |
| | clustering | clustering method | [83] [84] [67] [81] | group the input states to speed up matching |
| | inverse reinforcement learning | Convex Apprenticeship Learning Algorithm | [5] | learn weights of reward functions |
| | | entropy | [85] | compute feature weights by gradient-based optimization from data |
| | | probability | [86] | transfer examples to probability distributions |
| | neural network | matching current states and actions | [35] [87] [88] | evolve parameters for the specified model according to data |
| | | considering previous states | [89] [90] [91] [92] [93] [94] [95] | predict future trajectories according to previous states and current states |
| Rule-based | evolutionary algorithm | gene expression programming | [31] [32] | automate decision rule generation |
| | | self-learning gene expression programming | [96] | obtain general decision rule that can be adopted in scenes different from training scenes |
| | | genetic algorithm | [97] | learn crowd models from data according to a predefined meta-model |
| | decision tree | classification | [17] [33] | accelerate crowd simulation computation by classifying motion context |

according to empirical data. The walking speeds are generated from the speed distribution, and the cell size is determined by the maximal density. Hesham and Wainer [68] recently proposed an agent-based short-range collision-avoidance model for pedestrians in dense crowds. The authors showed their model's ability by tuning parameters of the model to reproduce emergent crowd phenomena that can match the real pedestrian trajectory data. Li et al. [98], in their review of cellular automata models, mentioned that, with the development of cognitive science and neural networks, how real data and cellular automata models are combined is important to the improvement of modeling accuracy. In fact, for models that only obtain several statistical parameters from the data mentioned above, how to make more use of data is also an urgent issue to improve them. In Section 3.4, more methods that use basic tuning combining other modeling techniques are introduced.

Some data-driven calibration methods adopt mathematical optimization techniques to tune the model parameters. For example, Scovanner et al. [16] presented a system that predicts pedestrian movements after being trained with observation data. Pedestrians' movements are formulated as sequences of continuous optimization. The choice of the next location at each time step is obtained through the optimization of a continuous function, which is formulated as an energy minimization problem. The standard gradient-based optimization technique is applied to optimize parameters so that the predicted tracks can match the observed ones. In Reference [34], the authors aim to model the following behaviors among pedestrians. Experiments were first conducted before calibration, where participants were asked to follow each other. The following behaviors were modeled by the least-squared method, which was then calibrated according to the experiment data. Another method proposed in Reference [69] focuses on modeling detailed interactions among humans when their trajectories are converging. Similar to Reference [34], experiments were first designed to generate real data, where two participants were asked to walk through an area simultaneously from different corners, and then the obtained data were used in the automatic calibration of the proposed model using the Maximum Likelihood Estimation technique. The article provides both calibrated and uncalibrated versions of the methods, and the calibrated model has better accuracy than the uncalibrated one. Yamaguchi et al. [70] proposed a crowd modeling method that uses an energy function that encodes a number of factors, including personal factors and social factors, to determine the next velocity of agents. There are eight parameters to be optimized, and a variant of the simplex algorithm is used to achieve this goal. In addition to optimizing parameters from data, the design of features used in the energy function is another key work. Vemula et al. [71] modeled an agent's velocity as a function of its goal and position. The interactions are modeled as a Gaussian Process, and the hyperparameters are learned from data by maximizing the log marginal likelihood of the Gaussian Process.

**Evolutionary algorithms (EA)** are commonly used methods for model calibration. In EA-based calibration methods, with data used for evaluation, a population of candidate solutions are evolved iteratively using nature-inspired operators, such as mutation and crossover. Helbing et al. [72] applied a simple evolutionary algorithm to the calibration of the social force model [99]. The real data are used to calculate the simulated error (fitness) of the evolved parameter set. The calibration model can be used in large scenes as long as important factors, such as maximum flow and density, are taken into account. To reduce the cost of calibration time resulting from simulation, surrogate models are also integrated into the framework [73]. Moussaïd et al. [74] calibrated parameters of the social force model by an evolutionary algorithm, so that simulated trajectories can be similar to observe ones. The three evolutionary methods mentioned before do not specify the type of evolutionary algorithm, but the following methods use specific evolutionary algorithms. Pellegrini et al. [11] designed a model for walking pedestrians with short-term predictions to avoid collisions before they occur, and it learns parameters from birds-eye-view data. The model has six parameters related to pedestrian behaviors, which need to be learned from the data. The local optimal parameter vectors resulting from **Genetic Algorithm (GA)** optimization with the lowest error are applied. Also, based on GA, Bera et al. [75] proposed a method to calibrate parameters of the RVO model with historical data. Then, the configured RVO model can be used to predict pedestrian trajectories. They further improved the method by using an adaptive particle filtering scheme, adding more applications and adding more datasets [76]. Johansson et al. [77] use the genetic algorithm to tune interaction parameters in social force model, to minimize the error of simulation. Except the genetic algorithm, the differential evolution is also a popular evolutionary algorithm. Tan et al. [78] proposed an agent-based data-driven model that focuses on the path planning layer of **origin/destination (OD)** popularities and route choice. The parameters are calibrated through the differential evolution genetic algorithm according to the density map obtained from the video. Zhong et al. [18] proposed a density-based evolutionary framework for automating the calibration of a crowd model. In the framework, a density-based matching scheme has been developed for fitness evaluation. The framework is improved by estimating how important the parameters are during calibration [79]. This estimation is accomplished by combining a sensitive analysis method and Powell's method to the evolution procedure.

### 3.2 Example-based Methods

Data-driven example-based methods extract information in the form of state-action pairs from data to construct models, which can then be used in simulation to determine agents' next actions according to their states. To achieve this determination, features need to be selected to represent states, and actions need to be extracted or learned from data during modeling.

Commonly, state-action pairs obtained from data are stored in a database. During the simulation, an agent's state is used as a key to search for a matching action as the value from the database. In Reference [43], a database of examples that describe the local spatiotemporal scenarios is first constructed. Then, agents search in the database for scenarios similar to their spatiotemporal states before taking actions. Ju et al. [80] proposed a modeling method that can generate an arbitrary number of agents according to input data. The approach collects the formation agent distributions and stores trajectory segments generated from videos in the database. During the simulation, the formation constraints are randomly set, and agents choose trajectory segments from the database to move under these constraints while avoiding collisions. This method is aimed at crowd animation, which is an application of crowd modeling, and real data are used to make the animation plausible. Another work presented by Toledo et al. [30] aims to generate steering behaviors of agents with a small amount of data. Because, in a real situation, people in similar states may act differently in the future, this method adopts a finite state machine combined with fuzzy logic to model agent's behavior so that there is uncertainty in agent behaviors. Though these agents act differently, how they act is still learned from data. The method combines data-driven steering and group social force. It learns from data the steering actions in different states that include velocities and closeness to goals and obstacles. During simulation, the system matches the agents' states with the knowledge base and gives the action vectors defining the next motions.

In some methods, the database only stores selected examples. Zhao et al. [14] recently proposed a role-dependent approach, which considers a pedestrian to be either a leader or a follower. The approach is specifically designed to simulate high-density crowds where pedestrians' movements are rather restricted compared to low-density crowds. The state-action pairs of leaders are stored in an example database organized by a k-d tree. Because, in a dense crowd, a pedestrian cares about others walking in the space along the direction of his preferred velocity, the vision area is represented by a series of strips. Figure 2 shows an example of a pedestrian's vision area. When a leader needs to decide where he heads to, the k-d tree is queried using the leader's states. The leader then obtains $K$ candidate actions to compute the final action.

Searching a huge database may cause inefficiency in application. Thus, there exist methods that first process the examples to solve the problems. In Reference [82], a data-driven spatiotemporal model for crowd animation generation is proposed. A time series of velocity fields is computed

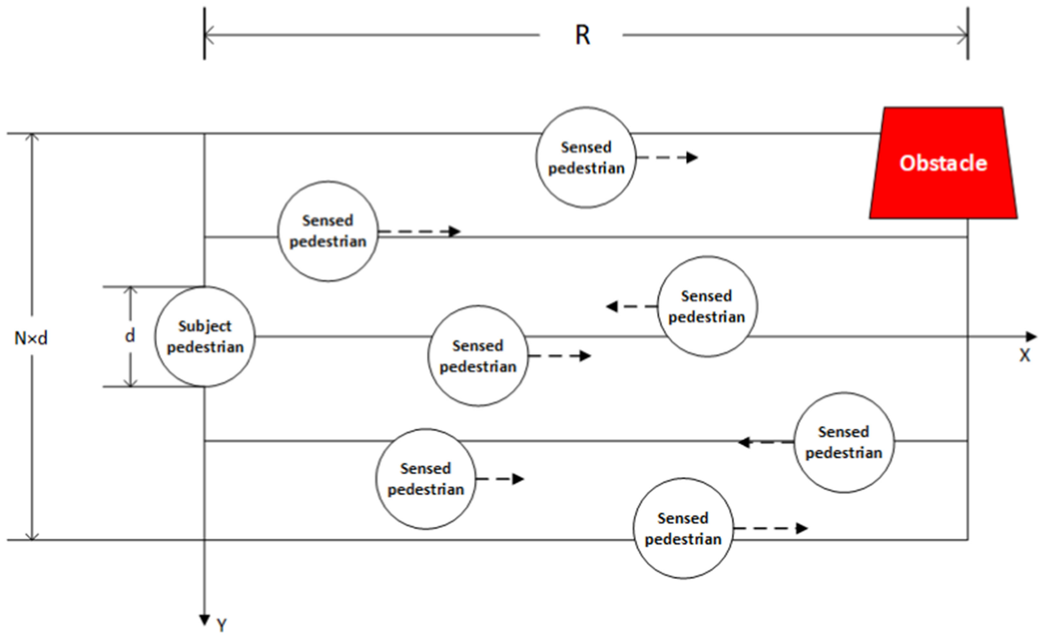


Fig. 2. An example of the pedestrian vision area in dense crowd [14].

with a motion estimation process. Then, instead of a large database, the velocity fields can be used to generate the entire animation. Lerner et al. [13] proposed a method that focuses on generating human behaviors rather than walking, such as chatting and looking around, at proper times. The method considers human actions to be motivated by stimuli, and it encodes the stimuli into examples. Trajectories are manually annotated with action tags, and stimuli-maps, which are density-based influence functions for determining action selection probability, are generated from the input videos to constrain the range of stimuli. During run-time, an agent's stimulus configuration determines the importance of each stimulus. Then, the importance is compared with the examples. Each action's probability is estimated, and the action is chosen according to this probability. Actions with high probabilities are more frequently chosen than those with low probabilities.

Another way to decrease simulation run-time overhead of example-based modeling approaches is the use of machine learning. Zhao et al. [83] presented a method that extracts examples from data and provides future behaviors for agents according to their states. It first groups agents' input states into different example clusters. An artificial neural network is then trained based on the clustering result to be a classifier. With clustering, the search space is reduced from all examples to a cluster during the simulation. Given the agent's state, consisting of the agent's path and the corresponding neighboring individuals in the past two seconds, a velocity vector indicating its position in the next time step can be selected from the cluster. Zhong et al. [84] proposed a method that learns spatial and temporal patterns from data. The **source regions (SR)** and **destination regions (DR)** are first initialized according to the scene. Then, the K-means algorithm is applied to cluster the trajectories into different groups according to their destinations. Each cluster can be seen as a velocity field of the corresponding DR and used to guide agents' navigation in the environment. The framework uses a dual-layer agent-based modeling architecture, where the bottom layer involves microscopic collision avoidance behaviors and the top layer uses the pattern learned as macro navigation. In this method, positions of agents can be considered as their states, and the velocities generated according to the corresponding velocity field will result in actions (i.e., the change of velocity).

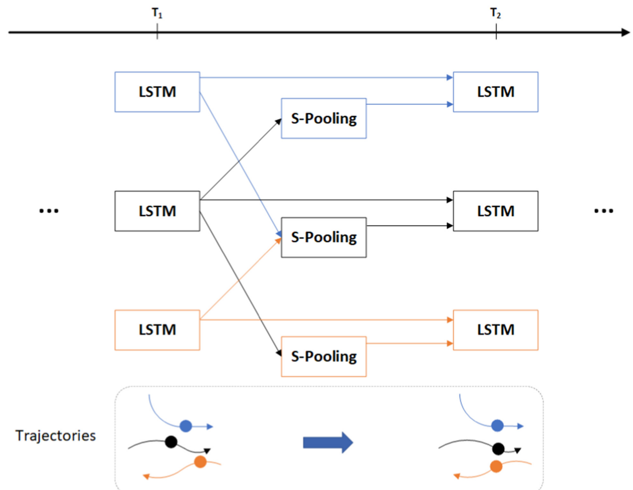


Fig. 3. Social-LSTM overview [89].

**Inverse reinforcement learning (IRL)** is a method for learning the reward function of reinforcement learning. In crowd modeling, the reward function is the combination of features used to evaluate the similarity between real data and simulated data. IRL for crowd modeling learns feature weights according to real-world data. Lee et al. [5] proposed a method that learns the weights by their new algorithm Convex Apprenticeship Learning Algorithm. The method learns the reward function as the behavior style in the reinforcement learning instead of specifying it. Kuderer et al. [85] modeled pedestrian behaviors based on the principle of maximum entropy. The problem is transferred to computing feature weights by gradient-based optimization. Another work [86] learns parameters of a mixture probability distribution that matches the observed behaviors in expectation. Because this method learns a probability distribution, the trajectories generated are diverse. It has less error than Reference [85], the social force model and RVO. In the Turing test, the percentage of behaviors induced by the proposed method qualified as human behaviors is closest to real human behaviors.

In addition to the techniques mentioned above, **neural networks (NNs)** have been adopted in example-based methods. For example, Wei et al. [35] proposed a method to generate an NN that can generate pedestrian behaviors directly with small datasets. The method mainly addresses the difficulties of realism, efficiency and maintainability. After a principal component analysis process that reduces the dimension of states, experiments are performed on state-action pairs to train the NN so that it can fit the data. The NN treated as a black-box tool reduces the difficulty of manual design. Song et al. [87] adopted a four-layer NN. To make it more adaptive to various scenarios, multiple scenario data are used to train the NN by normalizing relative positions among pedestrians, transferring velocity vectors to scalars, and incorporating more path planning information. The simulation results show that the proposed modeling method can generate crowds with more realistic density and smoother trajectories compared with the social force model. Ma et al. [88] rained an artificial neural network with data recorded by a camcorder to simulate pedestrian movements. An input state includes pedestrian current velocity, his interactions with other pedestrians and obstacles, and so on. The output is his response velocity.

Considering a path as a sequence of pedestrian positions, there exist crowd modeling methods using **long short-term memory (LSTM)** [100] for dynamic prediction. Alahi et al. [89] proposed the Social-LSTM for path prediction. Unlike traditional LSTM, Social-LSTM takes interactions among pedestrians into account by adding a social pooling layer. Each pedestrian considers his neighbors and his previous trajectory before determining the features of the next movement(i.e., velocity and gait). This consideration is achieved by connecting the LSTM of neighboring trajectories. Figure 3 shows how the Social-LSTM works. Fernando et al. [90] adopted LSTM combined with attention functions for pedestrian dynamics. The self-trajectory information is transferred to a soft attention context vector, while trajectories of neighbors are used to calculate hardwired attention context vectors. The generated vectors are then used to predict the subject pedestrian's future trajectory. Pfeiffer [91] pointed out that Social-LSTM only considers dynamic obstacles (i.e., other agents) and does not take static obstacles in the environment into account. Therefore, an LSTM method that considers both dynamic and static obstacles was proposed for pedestrian trajectory prediction. Bartoli et al. [92] proposed an LSTM able to combine human-human and human-space interactions, which can predict pedestrian trajectories in different places such as shopping malls and sidewalks.

Also considering previous movement information, several other NN methods are simply listed here. Yi et al. [93] adopted a deep convolutional neural network to predict future crowd motions from historical data. Zou et al. [94] proposed a framework of Social-Aware Generative Adversarial Imitation Learning to learn the decision process of humans from data using a **recurrent neural network (RNN)**. Vemula et al. [95] also proposed a method using an RNN to model trajectories. The interactions between each pedestrian are considered in this method.

### 3.3 Rule-based Methods

Data-driven rule methods learn behavior rules from data. The learned rules take environment features or individual properties as inputs and generate behaviors as outputs. The rule methods work in a similar manner to the example-based methods. Compared to example-based methods that memorize or summarize the corresponding action of each state with the support of a great amount of real data, rule-based methods try to reveal the principles of human behavior by considering physical, psychological and social factors. Rule-based methods are usually more general than example-based methods, because rule-based methods have a higher level of abstraction.

GA is a powerful tool for automatically generating rules. It treats each rule or model as an individual in the population, and after evolution using a fitness function, individuals in the population can have good performance. Zhong et al. [31] proposed an evolutionary framework, aiming to automate the process of generating decision rules for pedestrians. The method adopts a two-level model, where the lower level helps to avoid collisions by the social force model and the higher level guides agents' directions using a decision rule. The **gene expression programming (GEP)** is used to evolve the reward function as the rule in the higher level. The evaluation of the reward function is performed by comparing the simulated data with the video data. After this evaluation, agents are able to choose the best option with the largest reward. Namikoshi et al. [32] improved [31] by considering heterogeneous strategies in addition to homogeneous strategies. Each individual contains agents that are divided into groups. Agents in the same group use the same strategy that is evolved by GP. Each individual is represented by a tree whose children of the root are the groups of agents. Operators such as crossover and mutation are redefined so that the group members can change and groups can be integrated or divided.

Another evolutionary framework [96] that automates the generation of behavior rules solves the problem in a different manner. Instead of GEP, a new GP variant SL-GEP [101], which is easy to implement and requires fewer control parameters, is adopted. This method adopts a similar two-level model as described above and focuses on generating a navigation rule used in the top layer. The navigation rule takes features observed by the subject agent as inputs and outputs the intermediate direction the agent should follow. Compared with two non-data-driven navigation strategies, the proposed method can generate a crowd most similar to real data. Unlike many data-driven methods, the learned rule can be applied to scenes similar to the training scene and still obtain a plausible simulated crowd.

Keller et al. [97] proposed a method that uses GA to learn behavior models according to a predefined meta-model. The method first specifies the model space (that is, the meta-model) and then searches this space by GA. Previous works usually assume predefined model structures and elements. They use data to improve part of the models or to combine model elements [81, 102]. Different from these methods, the method in Reference [97] does not specify model templates in advance. Instead, it defines the general model space, which does not predefine specific elements of models. Actions are considered as the changes of agent properties. Each agent property can be modified by behaviors in the same behavior group. GA uses chromosomes to represent models that contain behavior groups and evolves the population to search for solutions. The generated model has good readability, and humans can understand it and adjust it by hand.

Boatright et al. [17] proposed a modeling method aiming to accelerate the computation of crowd simulation. The training data is generated by an oracle algorithm, which is based on an offline planner. The method splits the problem space into coarse features for the general world and fine features for other agents nearby. It uses a multilevel decision tree and classifies scenarios into different types of steering contexts according to the agent density, obstacle distributions and so on. Each steering context has a decision policy that is trained with extracted state-action pairs. The inputs of a policy are the fine features, and the outputs are the next footstep locations for agents. Each policy includes two boosted decision trees. This method treats crowd modeling as a classification problem instead of a regression problem. Kennedy et al. [33] proposed an intelligent assistance architecture for data-driven simulation. In the assistance architecture, high level descriptions summarizing the patterns or trends are generated for both simulation and real world data. Then, consistency checking is performed to determine whether the simulation states agree with the descriptions of data content, and the simulation model is adjusted to generate simulations similar to real world observations.

### 3.4 Other Methods

In this section, we list several crowd modeling methods that belong to no or more than one category mentioned above.

There exist modeling methods that aim to achieve crowd simulation visually similar to the input data in terms of macroscopic properties (e.g., density and flow). They usually learn the crowd from videos in a flow-based manner. Individual characteristics are neglected, so differences among pedestrians will not be found in the simulated crowd similar to those found in the real world. Lin et al. [103] proposed a method that learns the flow models from dynamic visual phenomena in videos by statistical tools. It can estimate the flow field from videos of crowds using statistical approaches. They also proposed another method [104] that models the persistent motion in dynamic scenes.

There are hybrid methods that learn more than one type of knowledge from data or use more than one way to learn. Liu et al. [66] proposed a method that improves the social force model for simulating crowd evacuation. It learns both trajectories and parameter values from real-world data. The method adopts the dual-layer structure mentioned before. In the top layer, agents select their goals according to a fitness function defined based on observation of the real data. Input features, such as the pedestrian number and the speed, are extracted from data. To simulate group behavior, leaders select routes according to trajectories (i.e., vectors of positions) learned from data. In the bottom layer, the group forces are added to the social force model, which can help

Table 3. Methods for Data-driven Crowd Model Validation

| Category | | Method | Reference | Description |
|---|---|---|---|---|
| metric-based | density | local density or local proximity | [36], [105] | a similarity function to measure the density similarity between simulated individuals and the real-world example. |
| | | fundamental diagram | [37] | a density-versus-speed diagram between simulated and real-world crowds to validate the effectiveness of long-range collision avoidance. |
| | | crowd density instantaneous fields | [106] | the instantaneous simulated results on different footbridges. |
| | | global density change | [107] | multiple measurement locations. |
| | | crowd in- and outflow | [108] | densities of pedestrians entering and leaving the scene at each moment. |
| | | the density distribution error | [18] | a density-based fitness validation module that measures the density distribution error in each time step |
| | trajectory | trajectory prediction error | [109], [110], [38], [89], [96] | use the absolute difference or the (root) mean square error to measure the trajectory differences between the simulated agents and the reference data at each time step. |
| | | generalized voronoi diagram | [111] | measure the topological equality and it is suitable in crowded scenarios. |
| | time | time difference | [15] | the time difference between the simulated crowd and real-world crowd passing through a certain region |
| | event | event classification error | [112], [113], [114] | mostly adopted in collision avoidance and abnormal behavior detection |
| | multi-metric | include density, the number of collisions, travel time, speed, speed change, travel distance, perpendicular deviation distance, angle change, biomechanical energy cost | [115], [116], [117] | methods (e.g., harmonic mean and expert experience) to intergrate different metrics |
| probability model-based | | the entropy method | [39], [118] | build a multidimensional Gaussian distribution based on the trajectory error vectors to represent a trajectory pattern and then measure the entropy of Gaussian probability models |
| | | the semantic method | [19], [20] | distribution differences between path patterns of trajectories |
| other | | global flow characteristics | [40] | applies machine learning techniques to extract features and measure similarity |
| | | crowd collectiveness | [119] | a degree measuring the pattern similarity of pedestrian behavior |

avoid collisions and generate plausible micro group behaviors. The learning approaches used in this method are basic tuning and databases. In terms of simulation results, compared to the original social force model, grouping behavior appears when using the proposed method. Kim et al. [67] proposed a method to generate plausible agent trajectories from data. The modeling method is used in interactive and adaptive crowd simulation by updating the initial position distribution and the movement flow model according to new observations provided by users. On the one hand, the initial position distribution is a mixture of multivariate Gaussian distributions and its parameters are learned from real trajectories. On the other hand, pedestrians are clustered according to their movement features, such as current positions, past velocities and preferred velocities. During the simulation, agents can be added to the scene according to the initial position distribution, and their preferred velocities can be updated by the features of the nearest cluster centroid. Therefore, this method learns both parameters and state-action pairs from data.

## 4 DATA-DRIVEN CROWD MODEL VALIDATION

Traditional crowd models are mainly validated based on the look and feel of the simulations that use these models, which requires a great amount of manual effort and is not suitable for crowd model validation for complex scenes. To facilitate automatic crowd modeling, automatic crowd

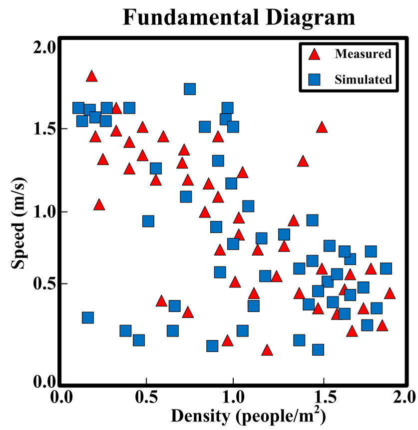


Fig. 4. An example of the fundamental diagram.

model validation has become a challenging and hot research topic in recent years. With the rapid development of computer vision techniques and the wide application of CCTV systems, many data-driven crowd model validation techniques that validate crowd models using crowd video data have been proposed. In this section, we discuss the existing data-driven crowd model validation techniques and these techniques are summarized in Table 3.

### 4.1 Metric-based Validation

**Metric-based data-driven crowd model validation (MDV)** uses a set of metrics to represent the attributes of simulated crowds and compares these metrics with those of real-world crowds to measure the performance of a model. Currently, it has become one of the most popular validation methods. A wide range of metrics and objectives are designed and commonly used in the literature, as shown in Table 3.

One of the most popular kinds of MDV methods is based on crowd density. Lerner et al. [36, 105] defined a similarity function to measure the similarity between simulated individuals and the real-world example individuals by using the local density or local proximity to represent the status of pedestrians. The distribution of similarity scores from the similarity function can further describe the differences between the simulated crowd and the real-world one. Moreover, in References [36, 105, 115], local density of pedestrians, number of collisions in simulation and travel time are also used as metrics to measure the differences between crowds. Singh et al. [115] proposed a scheme that considers a set of weighted metrics to measure the differences between crowd data. Golas et al. [37] adopted the fundamental diagram, a density-versus-speed diagram as shown in Figure 4, between simulated and real-world crowds to validate the effectiveness of long-range collision avoidance. Bruno et al. [106] used crowd density instantaneous fields to show the instantaneous simulated results on different footbridges. Pax et al. [107] also used global density change as a metric to validate the differences between simulated and real-world data. Feliciani et al. [108] used crowd in- and outflow as metrics, which are the densities of pedestrians entering and leaving the scene at each moment. A density-based fitness validation module that measures the density distribution error in each time step as the differences between two sets of crowd data, as shown in Figure 5, is also proposed in Reference [18].

Moreover, some methods use the error of certain aspects between the simulated and real-world data as the metrics of crowd models, such as the trajectory prediction error and the event

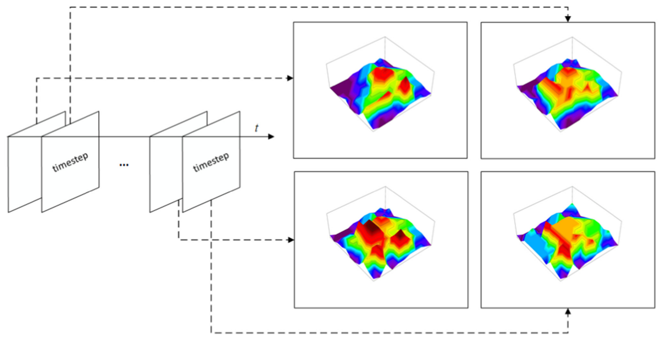


Fig. 5. An example of the macroscopic behavior represented by density distributions [18].

classification error. The trajectory prediction error MDV is mostly adopted in simulation applications. For example, Wolinski et al. [109, 110] and Zhong et al. [38] used the absolute difference between the simulated agents and the reference data at each time step to measure the performance of different algorithms. The (root-)mean-square error Equation (1) is also commonly used to measure the trajectory differences between crowds, and it is applied in References [89, 96]. In Equation (1), $i$ and $j$ are the individual index and the time step index, respectively, and $S$, $R$, and $N$ are the simulated data, reference data, and total number of time steps, respectively:

$$error_i = \frac{\sum_j^N (S_{ij} - R_{ij})^2}{N}. \tag{1}$$

In fact, both the absolute difference and the (root-)mean-square error ignore possible disturbances in motions of people. Even the same volunteer may walk differently in the same scenario because of uncertain factors, such as different emotions. Therefore, it may be impractical to make a crowd model that reproduces the perfect trajectories of volunteers. To overcome this problem, Stuvel et al. [111] and Basak et al. [15] utilized two more robust methods to evaluate the performance of crowds. Stuvel et al. [111] utilized a technique based on **Generalized Voronoi Diagram (GVD)** (i.e., the polygons with back straight lines in Figure 6) to measure the trajectory equality. As shown in Figure 6, the trajectories from the same agent (i.e., a circle) are regarded as topologically equal when these trajectories choose the same route between the same agents. For example, the red and blue trajectories in Figure 6(a) are regarded as the same. Otherwise, they are regarded as topologically unequal (i.e., the red and blue trajectories in Figure 6(b) are different). In addition, Basak et al. [15] used the time difference between the simulated crowd and real-world crowd passing through a certain region as the similarity metric concerning a crowd-swarming scenario.

The event classification error is a popular MDV method in collision avoidance and **abnormal behavior detection (ABD)**. For collision avoidance, an example work is proposed by Boatright et al. [112]. Because the motivation of their work is to predict the collision in a short time to improve the time efficiency of steering algorithms, the classification accuracy of collision error is designed as one of the metrics. For the ABD problems, the crowd models in ABD are expected to learn a pattern of abnormal and normal behavior to predict the event label of different video data, similar to the pattern recognition problems in computer vision. For example, Bera et al. [113] used the abnormal behavior prediction accuracy on all pedestrians in videos to quantify the performance of different algorithms. More metrics in ABD problems have been summarized in the survey by Sami et al. [114].

A set of statistical metrics have been proposed for crowd model validation [109, 110, 116, 117]. In Reference [116], four metrics (i.e., total evacuation time, average evacuation time, average

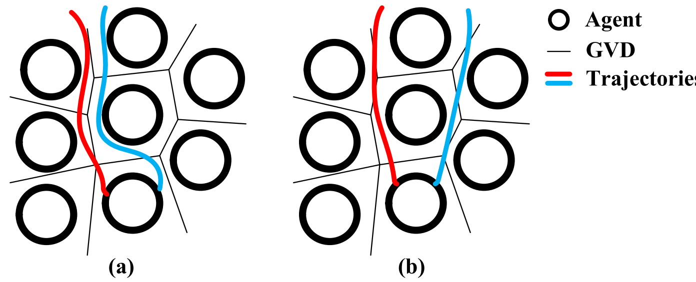


Fig. 6. An example of GVD trajectory validation method for a crowd. (a) two topologically equal trajectories; (b) two topologically unequal trajectories.

density, and average speed) are used, and a harmonic mean is calculated based on these four metrics. This harmonic mean is then ranked in ascending order and compared with the rank given by human experts. In Reference [117], seven metrics, including the average collision rate at each step, travel distance, perpendicular deviation distance, speed change, angle change, biomechanical energy cost and biomechanical energy cost in collision avoidance, are introduced and combined. The performances of the simulated crowds are compared based on the errors of these metrics between simulated crowds and real-world crowds, so researchers are able to comprehensively understand the characteristics of models.

### 4.2 Probability Model-based Validation

Besides MDV methods, **probability-based data-driven crowd model validation (PDV)** methods, the second main category of validation methods, try to establish a probability model to evaluate the performance of different models. One of the most representative PDV methods is the entropy method proposed by Guy et al. [39]. In their work, the error vectors (blue arrows) between the predicted movements by a certain crowd model (red line) and the corresponding real-world data (black line) are first collected, as shown in Figure 7. Then, the error vectors from the entire crowd are used to estimate the covariance matrix of a multidimensional Gaussian distribution, thereby formulating the error vectors of a crowd into a Gaussian distribution model. Finally, the magnitude of errors can thus be measured by the entropy of Gaussian probability models as Equation (2):

$$e(M) = \frac{1}{2} nlog((2\pi e)^d det(M)), \tag{2}$$

where $M$ is the covariance matrix of the Gaussian distribution model estimated based on the error vectors, $n$ is the number of agents in the crowd, and $d$ is the dimension of agents' states.

However, because the entropy method calculates the differences based on the error vectors, it cannot measure the difference between two behavior patterns of pedestrians with different moving directions. This entropy method is further improved by Karamouzas et al. [118]. In Reference [118], the validation is transformed into the per-time-step prediction tasks, and the validation method estimates a time-varying error for each agent independently to perform a per-agent entropy measure.

In addition to the entropy method, the semantic method, proposed by Wang et al. [19, 20], is another PDV method. Different from the entropy method, the semantic method can measure the similarity between two sets of trajectories in different locations, because the semantic method focuses on the differences between path patterns of trajectories instead of the trajectories themselves.

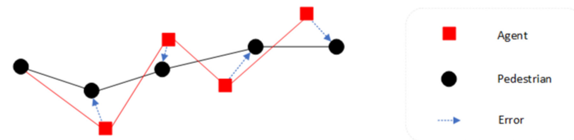


Fig. 7. Error vectors of the entropy method [39].

In principle, the semantic method assumes that there is a suite of basic path patterns (i.e., location-orientation pairs) in different trajectory datasets and that each trajectory can thereby be represented by different combinations of the weighted basic path patterns. With this hypothesis, the semantic method first uses multinomial distribution, the Dual Hierarchical Dirichlet Process, and Stochastic Variational Inference to extract the trending basic path patterns and estimate the corresponding posterior distribution of the trending path patterns based on the given training data. After that, the semantic method uses the extracted path patterns and their distribution to estimate the log likelihood of testing trajectories. In this manner, the behavior pattern similarity measurement is transformed into a predictive likelihood estimation problem as Equation (3). Given the training data and a part of the testing data, the target is to estimate the probability of generating the data coincident with the unseen (or incoming) testing data:

$$lik(B|A) = p(w^{ho}|A, w^{obs}), \tag{3}$$

where $A$ and $B$ are the training trajectory dataset and testing trajectory dataset, respectively, and $w^{obs}$ and $w^{ho}$ are the observed and held-out (i.e., unseen) trajectory data from testing dataset $B$. A higher likelihood of testing trajectories implies a higher similarity between training and testing trajectories, which means that the path pattern in training data is likely to generate trajectories similar to testing trajectories.

### 4.3 Other Validation Approaches

The validation methods that are not included in the above two types are categorized into other validation. These validation methods often have high flexibility to validate certain aspects of the proposed methods and good adaptation for different applications. For example, Musse et al. [40] proposed a model for quantitative comparison of global flow characteristics between the simulated crowd and the real one. In Reference [40], the validation model first clusters the raw trajectories into a high-level representation by **Mixture of Gaussian (MoG)**. Then, it builds a 4D histogram that describes the locations, directions and velocities for each cluster and uses a Bhattacharyya coefficient to measure the distance between clusters. The main difference of the method proposed in Reference [40] from MDV is that it applies several machine learning techniques, such as clustering and Gaussian kernel, to analyze the trajectory patterns and measures the similarity based on a newly defined histogram method instead of using statistical features (e.g., density and speed) as metrics directly. Additionally, in 2014, Zhou et al. [119] proposed a crowd collectiveness metric to measure the similarity between different crowds, where collectiveness is a degree measuring how much do pedestrians behave in the same pattern.

## 5 OPEN ISSUES AND FUTURE DIRECTIONS

In this section, we outline six open issues in the field of data-driven crowd modeling, as shown in Figure 8. The issues discussed below are not only common issues concerned by general simulation studies but also of significance in the context of crowd modeling.

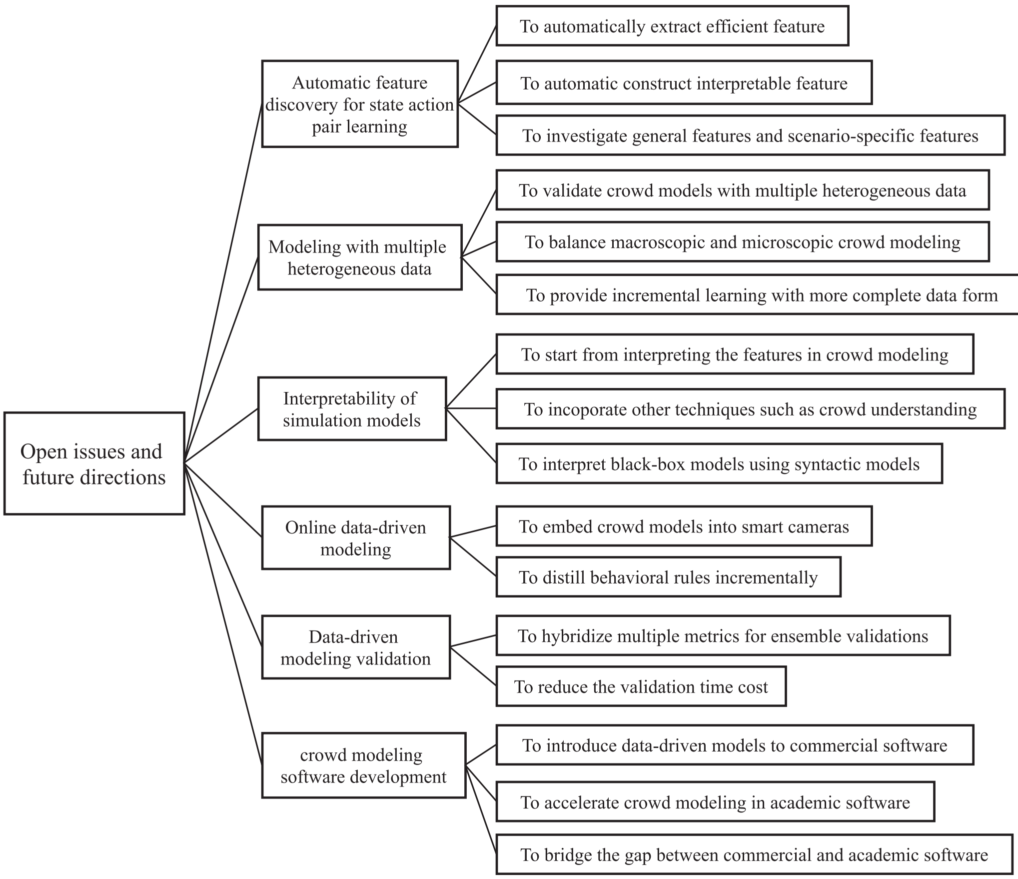


Fig. 8. An overall view of the open issues and the recommended future directions.

### 5.1 Automatic Feature Discovery for State-Action Pair Learning

Among state-action pair methods, one of the core issues is the definition of the states of agents. Most of the existing works [14, 30, 32, 83, 84, 96, 120–123] define states manually, which are usually represented by a set of features (e.g., the distances between an agent and obstacles, the density of the neighboring crowd around the agent, the angle between the destination and the agent's instant moving direction). These features are usually problem-specific and defined based on domain knowledge.

How to determine the most effective one(s) from a group of alternative features and how to reduce human efforts required to define these features are two issues that need to be addressed, especially in the context of crowd modeling. This is because, compared to other modeling tasks, the modeling of crowds is more complex and challenging to solve. For example, compared to traffic modeling, pedestrians share more diverse moving directions than vehicles, which move only along streets and roads. Thus, there may be more features involved in determining a pedestrian's future direction. Thus far, there has been no consensus regarding which features are the most useful for describing pedestrians' states. One reason is the diversity and complexity of simulation scenarios. For example, the corridor dataset[1] has a bottleneck shape that is long and narrow, and thus it

[1]http://www.asim.uni-wuppertal.de/datenbank/own-experiments/corridor/2d-bidirectional.html.

is important to consider the opposite crowd when predicting agents' next moves. Alternatively, when considering evacuation and rescue, the number of people escaping the area during a certain period of time is one of the primary concerns, and thus the instant people flow in different exits is weighted more than other features.

As for promising future research directions, the success of deep learning techniques makes it possible to extract high-level features when learning the moving patterns of pedestrians [89, 90, 124, 125]. With feature extraction and feature engineering techniques, it is possible to extract interacting features that collaboratively affect the route navigation process of human beings. These techniques will further reduce the number of features used in model training so that more efficient data-driven crowd models can be developed. In the meanwhile, features produced by extraction may be of high dimensionality and of low readability. In this case, the technique of feature construction can be used so that features that are not only effective but also in a human-understandable form can be generated. Machine learning techniques with good interpretability, such as decision trees and genetic programming, can be considered [126, 127]. More importantly, because different application scenarios demand different simulation purposes of crowd modeling, the generality of the already produced features may thus be reduced. For example, the physical shape of an indoor space, which may be a significant characteristic for emergency evacuation, may become trivial in sparse crowd modeling. Many implicit features need to be discovered in future works, and surveys need to be conducted to distinguish features that are general and compatible in cross-scenario crowd modeling from features that are highly specific.

### 5.2 Modeling with Multiple Heterogeneous Data

Training a data-driven model with one or a few data types may not be sufficient for accurate simulation or trajectory prediction. Multiple heterogeneous data can help solve the incompleteness problem of a single form of data [128]. Modeling with multiple heterogeneous data is essential for practical applications, especially for those complex prediction and decision-making processes. For example, researchers have applied multiple heterogeneous data to help diagnose Alzheimer's Disease [129] and cancers [130, 131].

In the context of crowd modeling, the heterogeneity of data refers to data from different types of data-gathering equipment, including but not limited to surveillance cameras, **infrared thermal imaging cameras (ITICs)**, various types of sensors, and the **Global Positioning System (GPS)**. Data heterogeneity is important in crowd modeling, not only because the movements of crowd are affected by various explicit and implicit factors, which require multiple feature representations, but also because there are different scales of modeling scenarios. For example, macroscopic crowd modeling usually regards pedestrians as a whole (e.g., fluid) and is more concerned with the group behaviors. Such characteristic makes it suitable to apply to simulate crowds in large areas. Thus, GPS may be a better choice for collecting data than surveillance cameras. Microscopic modeling, on the contrary, pays more attention to individual behaviors, and surveillance cameras, which are able to record position information of pedestrians more precisely and accurately, are preferable in this case. Moreover, the use of multiple heterogeneous data provides crowd modeling techniques with more feasibility. For example, a common surveillance camera may not be able to collect data in the nighttime due to the low light condition. ITICs can work well instead. Therefore, the combined use of both color surveillance cameras and ITICs provides crowd modeling with data collected all day long. Additionally, as indicated by the literature [32, 132], one of the major difficulties of crowd modeling comes from the heterogeneity of pedestrians' behaviors. With more diverse types of data, it should be more feasible to capture and model such heterogeneity. Moreover, crowd modeling relies on not only crowd data collected from cameras but also other data types, such as the statistical data from cognitive science. Such data is usually obtained through social investigations or questionnaires, and they are also regarded as heterogeneous. They are also important data, because pedestrians' behaviors may be affected not only by physical factors, such as the layout of the space and the density of the crowd, but also by social and psychological factors, such as pedestrians' genders, nationalities, ages, races, and so on [133, 134].

Thus far, little work has been done on incorporating crowd modeling with multiple heterogeneous data for better simulation results, and there is much room for research along this direction. For example, data heterogeneity can be incorporated with validation techniques to develop ensemble validation methods so that multiple aspects of a crowd model can be evaluated. Moreover, heterogeneity can also be used to enhance the generality of crowd models. For example, when online learning models are inputted with multiple heterogeneous data, they should be more robust to the spatiotemporal changes of human crowds. Additionally, data from surveillance cameras and from GPS can be combined together to balance the vividness of the macroscopic crowd flow with the precision of the microscopic individual behaviors. Last, the techniques of modeling with multiple heterogeneous data can also be incorporated with incremental learning so that data-driven crowd models can be trained with more complete data.

### 5.3 Interpretability of Simulation Models

In the fields of data mining and machine learning, the interpretability of a model refers to whether it can be explained with understandable terms and comprehended by humans [135]. Interpretability is important, because it helps build confidence in the safety, reliability, and fairness of models. Additionally, an interpretable model can be converted into understandable rules and theorems and further used by humans. Although black-box models have recently shown great capabilities when solving difficult tasks [94, 136–139], their incomprehensibility restricts their further performance. Machine learning models with interpretability [140–142], however, possess syntactic structures that help humans understand what principles the models follow when making decisions or predictions. Moreover, interpretable models can also be applied to explain those black-box models [141–144]. However, these models currently usually generate themselves with redundant structures, which decreases their own readability [145].

In the context of crowd modeling, interpretability is also essential. With well-performed machine learning models and effective features, the simulation results have been promising and close to reality. However, it is still hard to understand how these features interact with each other. As a cross-domain application, it is expected to not only obtain vivid simulation results but also explain the underlying reasons why pedestrians act in that manner. For example, in crowd modeling-assisted theme park design, designers may want to know not only which paths most tourists are likely to choose but also why they choose such paths so that specific services, stores, and entertainment can be placed alongside. Moreover, interpretability provides crowd models with not only better reliability but also a chance to discover the psychological and cognitive rules that humans obey in their subconscious. This is an essential issue, because it can promote the development of other involved subjects, such as cognitive science.

One way to enhance interpretability is to reduce the complexity of interpretable models. Efforts have been made toward models such as decision trees [146, 147] and genetic programming [148–151]. However, how to formally and mathematically define interpretability and how to quantitatively measure it in practice still remain open issues. To enhance the interpretability of decision trees and rule-based models, one of the most straightforward ways is to reduce all of the unnecessary components in the models according to the principle of Occam's Razor, so that the models can be made as simple as possible. For crowd modeling, it should be pointed out that interpretability is not an easy task. However, it is still worth making an attempt. With the development of feature extraction and construction techniques, it would be feasible to start interpreting a crowd model from its involved features. In the meanwhile, crowd understanding techniques [22, 59, 152] in the computer vision community can help to identify pedestrians' overall motion patterns, and thus combining them with crowd modeling can hopefully promote the interpretation progress. Moreover, black-box models and syntactic models can be combined together so that crowd models can take advantage of both the performance of the former and the interpretability of the latter.

### 5.4 Online Data-driven Modeling

With the rapid development of the Internet and digital and electronic technologies, various digital devices collect tons of data every second, and companies and governments accumulate enormous amounts of data every day. Big Data makes it feasible for online learning, which is also referred to as incremental learning. Online learning refers to learning from streaming data. Compared to classical batch training techniques in which machine learning models learn only once with a finite dataset, incremental learning has models that continue learning with consistently incoming data, thus providing models with better generalization ability. However, incrementally learning from dynamic data streams also introduces new challenges, one of which is the risk of concept drift. Concept drift refers to unpredictable changes of data distribution over time, which may mislead the well-trained learning model and degrade its performance. Moreover, incremental learning requires the ability to learn from a few samples of data and quickly convergence so that models can obtain better real-time responses.

In the context of data-driven crowd modeling, online learning is significant, not only because of the advantages of incremental learning itself but also because data-driven crowd modeling is intrinsically closely related to the online environment. First, human crowds are flowing flocks with diverse motion patterns. Habibi et al. [153] pointed out that pedestrian motion is more challenging to predict than vehicle routines given the unclear behavioral rules and the uncertainty of pedestrian motion. For classic batch learning, it is infeasible to implement a complete dataset to include all of these patterns [154], while for online learning, models only need to learn from what they have seen. Second, it is observed that crowds vary even in the same place with respect to different times [155]. For instance, crowds are denser during rush hours in metro stations, while railway stations obviously become crowded in public holidays. Online learning, in this case, adjusts crowd models to crowds with spatiotemporal changes so that the models can achieve better flexibility and generality. Moreover, in crowd modeling of related fields, such as robotics, robots are required to perceive and understand crowds' behavior, so that they will not offend humans during public services. Because human crowds move and change constantly, it is more suitable to have robots that learn online instead of pre-training them only once.

Pioneering efforts have been made toward online data-driven crowd modeling. Bera et al. [76] utilized an online object tracker to extract the first few frames of trajectories in videos and mixed the tracked trajectories with those predicted by a trained crowd simulation model to smoothen the predicted trajectories. By this means, they are able to apply the learned model to new scenarios without tedious manual trajectory annotation of new videos. The nonexistence of an absolutely comprehensive dataset motivated Habibi et al. [153] to adopt online learning to classify new pedestrian trajectories. These trajectories are first compared with previously seen ones. If a trajectory shares similar motion patterns, then it will be fused with the previous ones; otherwise, it will be added to capture new patterns. A sparse coding technique is then utilized for trajectory learning and prediction. The authors have proved the effectiveness of the proposed method over conventional offline ones.

Online data-driven crowd modeling is a promising future direction, and much work can be done. First, with the development of smart cameras, both data-driven crowd models and computer vision-based object tracking algorithms can be integrated into the camera through embedded techniques. Pedestrian trajectories can be constantly extracted from raw monitoring videos, and crowd behaviors can be incrementally learned by the model, with limited memory and computation power. Moreover, none of the aforementioned works focus on incrementally learning an underlying behavioral rule of crowds. Such a focus is essential, because online learning can verify and improve the generality of the distilled rules.

### 5.5 Data-driven Modeling Validation

Validation is an important topic, especially in the context of data-driven crowd modeling. Because crowd models involve a large number of parameters, validation results can return as a feedback to calibrate the values of these parameters. Meanwhile, because pedestrians' behaviors are so complex and affected by many factors including, but not limited to, physical ones, psychological ones and social ones, a single validation approach can hardly cover all of these factors; the incomplete coverage makes it possible to have one model that performs differently under two independent validations. For example, the collectiveness metric focuses on the macro level, while the trajectory error focuses on the micro level. Validating the simulation result by only one of them may emphasize either the group consistency or the individual vividness while ignoring the other. This validation is not comprehensive enough to reflect on the overall performance of the models. Therefore, to sufficiently compare the performances of different crowd models, it is better to adopt multiple validations, each focusing on a different aspect.

The road to accelerating validation for crowd modeling is still long, and much research is still to be performed. One promising future direction in designing automatic validation approaches is designing an assembled validation approach that hybridizes multiple validation metrics to generalize its applicability. Moreover, because the update of crowd models relies on the back-propagated validation results, and validation is usually a time-consuming task, an efficient validation technique can shorten the period from generating a parameter combination to acquiring the validation feedback. Therefore, reducing the validation time in model training is also a promising future research subject in data-driven model validation. Such future research makes sense for both academia and the industry. Currently, there are models capable of accelerating the validation process by shortening the time cost of simulation [41, 73, 156]. Surrogate-based models integrate a learning module to fit the underlying mapping from model parameters to validation results. Once such mapping has been learned, an approximate validation result can be predicted for a specific parameter combination, and the time-consuming regular simulation process can thus be skipped. Moreover, because agents always act simultaneously, microscopic crowd modeling possesses synchronism in its nature. High-performance computation tools and techniques can thus be added, such as GPU modules and various CPU parallel computation frameworks.

### 5.6 Crowd Modeling Software Development

Software for crowd modeling can be roughly divided into two types according to the purposes of usage, i.e., academic software for research purposes and commercial software for application development. Different purposes of usage result in different concerns when designing the software. For example, commercial crowd modeling software usually provides both good GUIs and simulation building blocks and prefer to use widely accepted models (e.g., the social force model). On the other hand, academic software is usually developed for scholarly communication. This development requires well-constructed modularity, as well as well-defined interfaces among different modules, so that others can further develop the software for their own purposes.

Table 4 shows ten popular general-purpose commercial software for crowd modeling. Among them, Viswalk, Simwalk, Massmotion, and Anylogic utilize the social force model for simulation;

Table 4. List of Popular Commercial Crowd Modeling and Simulation Platforms

| name | developer | applications | model | website |
|---|---|---|---|---|
| Pedestrian Dynamics | Incontrol Simulation | crowd simulation in infrastructure and building | cognitive-science-based vision and behavioral heuristics [167] | https://www.incontrolsim.com/software/pedestrian-dynamics/ |
| Viswalk | PTV AG | planning and optimization for large rallies | social force model | http://vision-traffic.ptvgroup.com/ |
| Simwalk | Savannah Simulation AG | passenger flow simulation at airports and railway terminals; Urban planning; crowd evacuation | potential filed algorithm based on social force model | http://www.simwalk.com/ |
| Massmotion | Oasys Ltd. | design, planning, and analysis of airports and stations | social force model | https://www.oasys-software.com/products/pedestrian-simulation/massmotion/ |
| Legion | Legion Ltd. | building design optimization | cellular automata | https://www.bentley.com/en/products/brands/legion |
| STEPS | Mott MacDonald Group Ltd. | designs of buildings, open areas, and infrastucture, city planning, evacuation | cellular automata | http://www.steps.mottmac.com/ |
| CAST Terminal | Airport Research Center GmbH | optimization of the designs of airport terminals andother passenger handling facilities | NOT FOUND | https://arc.de/cast-terminal-simulation/ |
| PedGo | TraffGo HT GmbH | crowd evacuation | cellular automata | https://traffgo-ht.com/en/pedestrians/products/pedgo/index.html |
| Exodus | University of Greenwich | crowd evacuation | rules or heuristics | http://fseg.gre.ac.uk/exodus/ |
| Anylogic | The Anylogic Company | multifunctional system simulations, with a pedestrian library; evacuation | social force model | https://www.anylogic.com/resources/libraries/pedestrian-library/ |

Legion, STEPS, and PedGo utilize cellular automata; Pedestrian Dynamics and Exodus utilize predefined heuristics. Additionally, Pedestrian Dynamics, Viswalk, Simwalk and Massmotion focus on similar applications (e.g., terminal design). This indicates the homogeneity of these commercial software, which may reduce their competitiveness in commercial markets. Moreover, although not being specifically designed for crowd modeling and thus not being included in the table, two popular game engines, Unity[2] and Unreal,[3] are also worth mentioning. These two engines have accumulated a large user base over the years. There are numerous third-party crowd simulation projects [157–159] developed in Unity and posted onto the asset store for paid downloads. Unreal integrates game-oriented crowd simulation interfaces, such as navigation modules and collision avoidance modules.

Table 5 lists seven popular academic frameworks that have been widely used among scientific publications [160–166]. Different from commercial products, which are closed source and authorized by licenses, academic software concerns about analyzable code and ensures repeatable exact computer experiments. Therefore, all of the frameworks listed in Table 5 are open source. Among them, frameworks such as Menge and Vadere integrate different widely accepted crowd models (the social force model, optimal steps model, gradient navigation model, rule-based model, etc.), which facilitates scientific comparison. These frameworks are generic enough to provide different functionalities within crowd modeling, such as motion planning, collision-avoidance and navigation and path planning. Simple but user-friendly GUIs, as well as diverse programming

[2] https://www.unity.com/.
[3] https://www.unrealengine.com/.

Table 5. List of Academic Crowd Modeling and Simulation Frameworks and Toolkits

| name | developer | type | description | language | website |
|---|---|---|---|---|---|
| Menge [168] | University of North Carolina–Chapel Hill | framework | facilitate crowd simulation research, development, and model comparison | C++ | http://gamma.cs.unc.edu/menge/ |
| Vadere [169] | Munich University of Applied Sciences | framework | integrate generic locomotion models for scientific comparison | Java | https://gitlab.lrz.de/vadere/ |
| Pedsim [170] | Christian Gloor | library & toolkit | crowd simulation; especially for large-scale scenarios | C++ | http://pedsim.silmaril.org/ |
| JuPedSim [171] | Forschungszentrum Julich | framework | facilite pedestrian dynamics researches; for evacuation; extensible to other areas | C++ | http://www.jupedsim.org/ |
| MoPlaT [172] | Nanyang Technological University | testbed | motion planning comparison; motion planning system analyses | Java | https://github.com/vaisaghvt/MoPlaT |
| Repast [173] | University of Chicago | platform | general purpose simulation and modeling | C# | https://repast.github.io/index.html |
| Netlogo [174] | Northwestern University | platform | general purpose simulation and modeling | Java | http://ccl.northwestern.edu/netlogo/ |

interfaces, are provided, which visualize the experimental results and encourage researchers to further develop the frameworks under specific research purposes. Noted that, although both Repast and Netlogo are platforms for general multi-agent systems, they have been adopted in the pedestrian simulation literature [163–166], and therefore we still include them in the academic software list. Moreover, there are also other tools involved in crowd simulation-related topics, such as Recast & Detour,[4] which is an open-source path-planning engine. Recast & Detour can be roughly divided into two parts, where the Recast module is for constructing a navigation mesh model, while Detour is for planning a humanoid path with respect to user-customized requirements.

Much work can be done to improve the quality of both commercial and academic software in the context of data-driven crowd modeling. For commercial software, all of the models mentioned in Table 4 are not data-driven, which means that the software lacks data-driven capabilities (e.g., parameters of the social force model in commercial software are usually fixed and cannot be automatically calibrated with data) or support for data-driven model development. Compared with traditional models, such as cellular automata and the social force model, data-driven models can achieve higher accuracy [89, 96] and may be more qualified for a diverse range of crowd modeling applications. Therefore, adaptive parameter calibration techniques can be added so that the softwares can adjust themselves to different simulation scenarios and be free from users' experience of parameter settings. Additionally, more advanced models (e.g., genetic programming-based models and deep learning-based models) can be introduced into the software so that users can have more choices according to different requirements in different simulation tasks. For academic software, because data-driven crowd modeling usually involves many time-consuming simulations, designing efficient parallel data-driven crowd modeling software is a promising research topic. Last, bridging the gap between the academic research of effective data-driven crowd modeling techniques and the integration of advanced models into commercial software is still an open issue.

## 6 CONCLUSIONS

Data-driven crowd modeling techniques model pedestrian behaviors according to real-world data. Compared to traditional crowd modeling techniques, data-driven ones are able to generate more

[4] https://masagroup.github.io/recastdetour/.

realistic crowds when they are applied in crowd simulation. During the past few years, a great number of data-driven crowd modeling techniques have emerged. In this article, we first list open-source crowd datasets that can be used in data-driven crowd modeling. Then, the state-of-the-art data-driven crowd modeling methods are introduced and classified into four types according to what they learn from data. For each type, how these methods learn is further discussed. Next, we discuss crowd simulation validation methods, which play an important role in data-driven modeling. Validation methods in this survey include metric-based validation methods, probability model-based validation methods and others. In spite of the great number of related works, there is still room for development of data-driven crowd modeling. First, because feature selection plays a fundamental role in state-action pair learning and, in most works, features are selected manually, automatic feature discovery is a promising way to improve the learning efficiency. Second, heterogeneous data from various sources may help to improve both the accuracy and generality of the model. Third, interpretability of the models should be improved so that users can have confidence in them and modify them conveniently according to their needs. Fourth, with the rapid development of surveillance camera technology and easy availability of video data, online data-driven modeling techniques need to be developed. Fifth, data-driven simulation validation can be developed, because it is usually a difficult task to choose a proper validation method. Last, bridging the gap between the academic research and the integration of advanced data-driven models into commercial software is still an open issue.

## REFERENCES


[1] Yang Cong, Junsong Yuan, and Ji Liu. 2013. Abnormal event detection in crowded scenes using sparse representation. *Pattern Recogn.* 46, 7 (2013), 1851–1864.

[2] Ramin Mehran, Alexis Oyama, and Mubarak Shah. 2009. Abnormal crowd behavior detection using social force model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* IEEE, 935–942.

[3] Q. H. Nguyen, Frederic D. McKenzie, and Mikel D. Petty. 2005. Crowd behavior cognitive model architecture design. In *Proceedings of the Conference on Behavior Representation in Modeling and Simulation (BRIMS'05).* 55–64.

[4] Shangping Ting and Suiping Zhou. 2008. Snap: A time critical decision-making framework for MOUT simulations. *Comput. Animat. Virt. Worlds* 19, 3–4 (2008), 505–514.

[5] Seong Jae Lee and Zoran Popović. 2010. Learning behavior styles with inverse reinforcement learning. *ACM Trans. Graph.* 29, 4 (2010), 122.

[6] Sean Curtis, Jamie Snape, and Dinesh Manocha. 2012. Way portals: Efficient multi-agent navigation with line-segment goals. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games.* ACM, 15–22.

[7] Paulo Knob, Marlon Alcantara, Estêvão Testa, Rodolfo Favaretto, Gabriel Lima, Leandro Dihl, and Soraia Raupp Musse. 2018. Generating background NPCs motion and grouping behavior based on real video sequences. *Entertain. Comput.* 27 (2018), 179–187.

[8] Kevin Jordao, Panayiotis Charalambous, Marc Christie, Julien Pettré, and Marie-Paule Cani. 2015. Crowd art: Density and flow-based crowd motion design. In *Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games.* ACM, 167–176.

[9] Sixuan Wang and Gabriel Wainer. 2015. A simulation as a service methodology with application for crowd modeling, simulation and visualization. *Simulation* 91, 1 (2015), 71–95.

[10] George Kouskoulis, Ioanna Spyropoulou, and Constantinos Antoniou. 2018. Pedestrian simulation: Theoretical models vs. data driven techniques. *Int. J. Transport. Sci. Technol.* 7, 4 (2018), 241–253.

[11] S. Pellegrini, A. Ess, K. Schindler, and L. van Gool. 2009. You'll never walk alone: Modeling social behavior for multi-target tracking. In *Proceedings of the IEEE 12th International Conference on Computer Vision.* 261–268.

[12] Daniel R. Parisi, Pablo A. Negri, and Luciana Bruno. 2016. Experimental characterization of collision avoidance in pedestrian dynamics. *Phys. Rev. E* 94, 2 (2016), 022318.

[13] Alon Lerner, Eitan Fitusi, Yiorgos Chrysanthou, and Daniel Cohen-Or. 2009. Fitting behaviors to pedestrian simulations. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation.* 199–208.

[14] Mingbi Zhao, Jinghui Zhong, and Wentong Cai. 2018. A role-dependent data-driven approach for high-density crowd behavior modeling. *ACM Trans. Model. Comput. Simul.* 28, 4 (2018), 28.

[15] Ahmet Eren Ba sak, Uğur Güdükbay, and Funda Durupınar. 2018. Using real life incidents for creating realistic virtual crowds with data-driven emotion contagion. *Comput. Graph.* 72 (2018), 70–81.

[16] Paul Scovanner and Marshall F. Tappen. 2009. Learning pedestrian dynamics from the real world. In *Proceedings of the IEEE 12th International Conference on Computer Vision*. 381–388.
[17] Cory D. Boatright, Mubbasir Kapadia, Jennie M. Shapira, and Norman I. Badler. 2013. Context-sensitive data-driven crowd simulation. In *Proceedings of the 12th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry*. 51–56.
[18] Jinghui Zhong, Nan Hu, Wentong Cai, Michael Lees, and Linbo Luo. 2015. Density-based evolutionary framework for crowd model calibration. *J. Comput. Sci.* 6 (2015), 11–22.
[19] He Wang, Jan Ondrej, and Carol O'Sullivan. 2016. Path patterns: Analyzing and comparing real and simulated crowds. In *Proceedings of the 20th ACM Siggraph Symposium on Interactive 3D Graphics and Games (I3D'16)*. 49–57.
[20] H. Wang, J. Ondrej, and C. O'Sullivan. 2017. Trending paths: A new semantic-level metric for comparing simulated and real crowd data. *IEEE Trans. Vis. Comput. Graph.* 23, 5 (2017), 1454–1464. https://doi.org/10.1109/TVCG.2016.2642963
[21] Teng Li, Huan Chang, Meng Wang, Bingbing Ni, Richang Hong, and Shuicheng Yan. 2015. Crowded scene analysis: A survey. *IEEE Trans. Circ. Syst. Video Technol.* 25, 3 (2015), 367–386.
[22] Jason M. Grant and Patrick J. Flynn. 2017. Crowd scene understanding from video: A survey. *ACM Trans. Multimedia Comput. Commun. Appl.* 13, 2 (2017), 19.
[23] Hairong Dong, Min Zhou, Qianling Wang, Xiaoxia Yang, and Fei-Yue Wang. 2020. State-of-the-art pedestrian and evacuation dynamics. *IEEE Trans. Intell. Transport. Syst.* 21, 5 (2020), 1849–1866.
[24] A. Adam, E. Rivlin, I. Shimshoni, and D. Reinitz. 2008. Robust real-time unusual event detection using multiple fixed-location monitors. *IEEE Trans. Pattern Anal. Mach. Intell.* 30, 3 (2008), 555–560. https://doi.org/10.1109/tpami.2007.70825
[25] James Ferryman and Anna Louise Ellis. 2010. PETS2010: Dataset and challenge. In *Proceedings of the 7th IEEE International Conference on Advanced Video and Signal-based Surveillance*. 143–150.
[26] Ke Chen, Chen Change Loy, Shaogang Gong, and Tao Xiang. 2012. *Feature Mining for Localised Crowd Counting*. https://doi.org/10.5244/c.26.21
[27] Xiao Song, Jinghan Sun, Jing Liu, Kai Chen, and Hongnan Xie. [n.d.]. OpenPTDS dataset: Pedestrian trajectories in crowded scenarios. In *Methods and Applications for Modeling and Simulation of Complex Systems*. Springer, Singapore, 296–305.
[28] Bastian Leibe, Konrad Schindler, and Luc Van Gool. 2007. Coupled detection and trajectory estimation for multi-object tracking. In *Proceedings of the IEEE 11th International Conference on Computer Vision*. IEEE, 1–8.
[29] Yijiang Zhang, Julien Pettré, Qunsheng Peng, and Stéphane Donikian. 2009. Data-based steering of virtual human using a velocity-space approach. In *Proceedings of the 2nd International Workshop on Motion in Games (MIG'09)*. 170–181.
[30] Leonel Toledo, Ivan Rivalcoba, and Isaac Rudomin. 2018. Fuzzy and data-driven urban crowds. In *Proceedings of the International Conference on Computational Science*. 280–290.
[31] Jinghui Zhong, Linbo Luo, Wentong Cai, and Michael Lees. 2014. Automatic rule identification for agent-based crowd models through gene expression programming. In *Proceedings of the International Conference on Autonomous Agents and Multi-agent Systems*. 1125–1132.
[32] Keiichi Namikoshi and Sachiyo Arai. 2018. Estimation of the heterogeneous strategies from action log. In *Proceedings of the Genetic and Evolutionary Computation Conference*. 1310–1317.
[33] Catriona Kennedy, Georgios Theodoropoulos, Volker Sorge, Edward Ferrari, Peter Lee, and Chris Skelcher. 2007. AIMSS: An architecture for data driven simulations in the social sciences. In *Proceedings of the International Conference on Computational Science (ICCS'07)*, Yong Shi, Geert Dick van Albada, Jack Dongarra, and Peter M. A. Sloot (Eds.). Springer, Berlin, 1098–1105.
[34] Samuel Lemercier, Asja Jelic, Richard Kulpa, Jiale Hua, Jérôme Fehrenbach, Pierre Degond, Cécile Appert-Rolland, Stéphane Donikian, and Julien Pettré. 2012. Realistic following behaviors for crowd simulation. *Comput. Graph. Forum* 31, 2 (2012), 489–498.
[35] Xiang Wei, Wei Lu, Lili Zhu, and Weiwei Xing. 2018. Learning motion rules from real data: Neural network for crowd simulation. *Neurocomputing* 310 (2018), 125–134.
[36] Alon Lerner, Yiorgos Chrysanthou, Ariel Shamir, and Daniel Cohen-Or. 2009. Data driven evaluation of crowds. In *Proceedings of the International Workshop on Motion in Games*. 75–83.
[37] A. Golas, R. Narain, S. Curtis, and M. C. Lin. 2014. Hybrid long-range collision avoidance for crowd simulation. *IEEE Trans. Visual. Comput. Graph.* 20, 7 (2014), 1022–1034.
[38] J. Zhong, W. Cai, L. Luo, and M. Zhao. 2016. Learning behavior patterns from video for agent-based crowd modeling and simulation. *Auton. Agents Multi-agent Syst.* 30, 5 (2016), 990–1019. https://doi.org/10.1007/s10458-016-9334-8
[39] Stephen J. Guy, Jur Van Den Berg, Wenxi Liu, Rynson Lau, Ming Lin, and Dinesh Manocha. 2012. A statistical similarity measure for aggregate crowd dynamics. *ACM Trans. Graph.* 31, 6 (2012), 190.

[40] S. R. Musse, V. J. Cassol, and C. R. Jung. 2012. Towards a quantitative approach for comparing crowds. *Comput. Animat. Virt. Worlds* 23, 1 (2012), 49–57. https://doi.org/10.1002/cav.1423

[41] A. Tavcar and M. Gams. 2018. Surrogate-agent modeling for improved training. *Eng. Appl. Artific. Intell.* 74 (2018), 280–293.

[42] Antoni B. Chan and Nuno Vasconcelos. 2008. Modeling, clustering, and segmenting video with mixtures of dynamic textures. *IEEE Trans. Pattern Anal. Mach. Intell.* 30, 5 (2008), 909–926.

[43] Alon Lerner, Yiorgos Chrysanthou, and Dani Lischinski. 2007. Crowds by example. *Comput. Graph. Forum* 26, 3 (2007), 655–664. https://doi.org/10.1111/j.1467-8659.2007.01089.x

[44] Saad Ali and Mubarak Shah. 2008. Floor fields for tracking in high density crowd scenes. In *Proceedings of the European Conference on Computer Vision*. Springer, 1–14.

[45] Mikel Rodriguez, Josef Sivic, Ivan Laptev, and Jean Yves Audibert. 2011. Data-driven crowd analysis in videos. In *Proceedings of the International Conference on Computer Vision*. 1235–1242.

[46] B. L. Zhou, X. O. Tang, and X. G. Wang. 2015. Learning collective crowd behaviors with dynamic pedestrian-agents. *Int. J. Comput. Vis.* 111, 1 (2015), 50–68. https://doi.org/10.1007/s11263-014-0735-3

[47] Saad Ali and Mubarak Shah. 2007. A lagrangian particle dynamics approach for crowd flow segmentation and stability analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 1–6.

[48] Bolei Zhou, Xiaoou Tang, and Xiaogang Wang. 2013. Measuring crowd collectiveness. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3049–3056.

[49] C. Zhang, K. Kang, H. Li, X. Wang, R. Xie, and X. Yang. 2016. Data-driven crowd understanding: A baseline for a large-scale crowd dataset. *IEEE Trans. Multimedia* 18, 6 (2016), 1048–1061.

[50] Alexandre Robicquet, Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. 2016. Learning social etiquette: Human trajectory understanding in crowded scenes. In *Proceedings of the European Conference on Computer Vision (ECCV'16)*. Springer, 549–565.

[51] R. Mazzon and A. Cavallaro. 2013. Multi-camera tracking using a multi-goal social force model. *Neurocomputing* 100 (2013), 41–50. https://doi.org/10.1016/j.neucom.2011.09.038

[52] Stefania Bandini, Andrea Gorrini, and Giuseppe Vizzari. 2014. Towards an integrated approach to crowd analysis and crowd synthesis: A case study and first results. *Pattern Recogn. Lett.* 44 (2014), 16–29. https://doi.org/10.1016/j.patrec.2013.10.003

[53] Anh Tuan Nghiem, Francois Bremond, Monique Thonnat, and Valery Valentin. 2007. ETISEO, performance evaluation for video surveillance systems. In *Proceedings of the IEEE Conference on Advanced Video and Signal-based Surveillance*. 476–481.

[54] X. Zhu, C. Loy, and S. Gong. 2016. Learning from multiple sources for video summarisation. *Int. J. Comput. Vision* 117, 3 (2016), 247–268. https://doi.org/10.1007/s11263-015-0864-3

[55] Barbara Majecka. 2009. *Statistical models of pedestrian behaviour in the forum*. Master's Thesis, School of Informatics, University of Edinburgh (2009).

[56] Edwin Richard Galea, Steven Deere, Robert Brown, and Lazaros Filippidis. 2013. An experimental validation of an evacuation model using data sets generated from two large passenger ships. *J. Ship Res.* 57, 3 (2013), 155–170.

[57] R. Brown, E. R. Galea, S. Deere, and L. Filippidis. 2013. Passenger response time data-sets for large passenger ferries and cruise ships derived from sea trials. *Trans. Roy. Inst. Naval Arch. Int. J. Maritime Eng.* 155, A1 (2013), 33–48.

[58] S. J. Deere, E. R. Galea, L. Filippidis, and R. Brown. 2012. Data collection methodologies used in the SAFEGUARD project to collect human factors data. In *Proceedings of the RINA SAFEGUARD Passenger Evacuation Seminar*.

[59] C. Zhang, K. Kang, H. Li, X. Wang, R. Xie, and X. Yang. 2016. Data-driven crowd understanding: A baseline for a large-scale crowd dataset. *IEEE Trans. Multimedia* 18, 6 (2016), 1048–1061.

[60] Suiping Zhou, Dan Chen, Wentong Cai, Linbo Luo, Malcolm Yoke Hean Low, Feng Tian, Victor Suhan Tay, Darren Wee Sze Ong, and Benjamin D Hamilton. 2010. Crowd modeling and simulation technologies. *ACM Trans. Model. Comput. Simul.* 20, 4 (2010), 20.

[61] Gianluca Antonini, Michel Bierlaire, and Mats Weber. 2006. Discrete choice models of pedestrian walking behavior. *Transport. Res. Part B: Methodol.* 40, 8 (2006), 667–687.

[62] Sébastien Paris, Julien Pettré, and Stéphane Donikian. 2007. Pedestrian reactive navigation for crowd simulation: A predictive approach. In *Proceedings of the Computer Graphics Forum*, Vol. 26. 665–674.

[63] Serge P. Hoogendoorn and Winnie Daamen. 2007. Microscopic calibration and validation of pedestrian models: Cross-comparison of models using experimental data. In *Proceedings of the Conference on Traffic and Granular Flow*. Springer, 329–340.

[64] David C. Brogan and Nicholas L. Johnson. 2003. Realistic human walking paths. In *Proceedings of the 11th IEEE International Workshop on Program Comprehension*. 94–101.

[65] Hubert Klüpfel. 2003. A cellular automaton model for crowd movement and egress simulation. https://nbn-resolving.org/urn:nbn:de:hbz:464-duett-08012003-0925403.

[66] Baoxi Liu, Hong Liu, Hao Zhang, and Xin Qin. 2018. A social force evacuation model driven by video data. *Simul. Model. Pract. Theory* 84 (2018), 190–203.
[67] Sujeong Kim, Aniket Bera, Andrew Best, Rohan Chabra, and Dinesh Manocha. 2016. Interactive and adaptive data-driven crowd simulation. In *Proceedings of the IEEE Conference on Virtual Reality (VR'16)*. 29–38.
[68] Omar Hesham and Gabriel Wainer. 2021. Advanced models for centroidal particle dynamics: Short-range collision avoidance in dense crowds. *Simulation* 97 (2021), 8. https://journals.sagepub.com/doi/10.1177/00375497211003126.
[69] Julien Pettré, Jan Ondřej, Anne-Hélène Olivier, Armel Cretual, and Stéphane Donikian. 2009. Experiment-based modeling, simulation and validation of interactions between virtual walkers. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 189–198.
[70] Kota Yamaguchi, Alexander C. Berg, Luis E. Ortiz, and Tamara L. Berg. 2011. Who are you with and where are you going. In *Proceedings of the 24th IEEE Conference on Computer Vision and Pattern Recognition (CVPR'11)*. 1345–1352.
[71] Anirudh Vemula, Katharina Muelling, and Jean Oh. 2017. Modeling cooperative navigation in dense human crowds. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'17)*. IEEE, 1685–1692.
[72] D. Helbing, A. Johansson, and P. K. Shukla. 2007. Specification of a microscopic pedestrian model by evolutionary adjustment to video tracking data. *Adv. Complex Syst.* 10 (2007), 271–288.
[73] Wenchao Yi, Jinghui Zhong, Singkuang Tan, Wentong Cai, and Nan Hu. 2017. Surrogate assisted calibration framework for crowd model calibration. In *Proceedings of the Winter Simulation Conference*. IEEE, 91.
[74] Mehdi Moussaïd, Dirk Helbing, Simon Garnier, Anders Johansson, Maud Combe, and Guy Theraulaz. 2009. Experimental study of the behavioural mechanisms underlying self-organization in human crowds. *Proc. Roy. Soc. B: Biol. Sci.* 276, 1668 (2009), 2755–2762.
[75] Aniket Bera, Sujeong Kim, and Dinesh Manocha. 2015. Efficient trajectory extraction and parameter learning for data-driven crowd simulation. In *Proceedings of the 41st Graphics Interface Conference*. Canadian Information Processing Society, 65–72.
[76] Aniket Bera, Sujeong Kim, and Dinesh Manocha. 2016. Online parameter learning for data-driven crowd simulation and content generation. *Comput. Graph.* 55 (2016), 68–79.
[77] Anders Johansson. 2009. Data-driven modeling of pedestrian crowds. https://core.ac.uk/reader/236365400.
[78] Sing Kuang Tan, Nan Hu, and Wentong Cai. 2018. Data-driven agent-based simulation for pedestrian capacity analysis. In *Proceedings of the International Conference on Computational Science*. 103–116.
[79] Jinghui Zhong and Wentong Cai. 2015. Differential evolution with sensitivity analysis and the Powell's method for crowd model calibration. *J. Comput. Sci.* 9 (2015), 26–32.
[80] Eunjung Ju, Myung Geol Choi, Minji Park, Jehee Lee, Kang Hoon Lee, and Shigeo Takahashi. 2010. Morphable crowds. In *ACM Transactions on Graphics*, Vol. 29. 140.
[81] Kang Hoon Lee, Myung Geol Choi, Qyoun Hong, and Jehee Lee. 2007. Group behavior from video: A data-driven approach to crowd simulation. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 109–118.
[82] Nicolas Courty and Thomas Corpetti. 2007. Crowd motion capture. *Comput. Animat. Virt. Worlds* 18, 4-5 (2007), 361–370.
[83] Mingbi Zhao, Stephen John Turner, and Wentong Cai. 2013. A data-driven crowd simulation model based on clustering and classification. In *Proceedings of the IEEE/ACM 17th International Symposium on Distributed Simulation and Real Time Applications*. 125–134.
[84] Jinghui Zhong, Wentong Cai, Linbo Luo, and Haiyan Yin. 2015. Learning behavior patterns from video: A data-driven framework for agent-based crowd modeling. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems*. 801–809.
[85] Markus Kuderer, Henrik Kretzschmar, Christoph Sprunk, and Wolfram Burgard. 2012. Feature-based prediction of trajectories for socially compliant navigation. In *Proceedings of the Conference on Robotics: Science and Systems*, Vol. 8.
[86] Henrik Kretzschmar, Markus Kuderer, and Wolfram Burgard. 2014. Learning to predict trajectories of cooperatively navigating agents. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'14)*. IEEE, 4015–4020.
[87] Xiao Song, Daolin Han, Jinghan Sun, and Zenghui Zhang. 2018. A data-driven neural network approach to simulate pedestrian movement. *Physica A: Stat. Mech. Appl.* 509 (2018), 827–844.
[88] Yi Ma, Eric Wai Ming Lee, and Richard Kwok Kit Yuen. 2016. An artificial intelligence-based approach for simulating pedestrian movement. *IEEE Trans. Intell. Transport. Syst.* 17, 11 (2016), 3159–3170.
[89] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. 2016. Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 961–971.
[90] Tharindu Fernando, Simon Denman, Sridha Sridharan, and Clinton Fookes. 2018. Soft + hardwired attention: An lstm framework for human trajectory prediction and abnormal event detection. *Neural Netw.* 108 (2018), 466–478.

[91] Mark Pfeiffer. 2018. Learning to navigate: Data-driven motion planning for autonomous ground robots. https://www.research-collection.ethz.ch/handle/20.500.11850/324253.
[92] F. Bartoli, G. Lisanti, L. Ballan, and A. Del Bimbo. 2018. Context-aware trajectory prediction. In *Proceedings of the 24th International Conference on Pattern Recognition (ICPR'18)*. 1941–1946. https://doi.org/10.1109/ICPR.2018.8545447
[93] Shuai Yi, Hongsheng Li, and Xiaogang Wang. 2016. Pedestrian behavior understanding and prediction with deep neural networks. In *Proceedings of the European Conference on Computer Vision (ECCV'16)*. Springer, 263–279.
[94] Haosheng Zou, Hang Su, Shihong Song, and Jun Zhu. 2018. Understanding human behaviors in crowds by imitating the decision-making process. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence.*
[95] A. Vemula, K. Muelling, and J. Oh. 2018. Social attention: Modeling attention in human crowds. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'18)*. 4601–4607. https://doi.org/10.1109/ICRA.2018.8460504
[96] Jinghui Zhong, Wentong Cai, Michael Lees, and Linbo Luo. 2017. Automatic model construction for the behavior of human crowds. *Appl. Soft Comput.* 56 (2017), 368–378.
[97] Nicholas Keller and Xiaolin Hu. 2019. Towards data-driven simulation modeling for mobile agent-based systems. *ACM Trans. Model. Comput. Simul.* 29, 1 (2019), 1.
[98] Yang Li, Maoyin Chen, Zhan Dou, Xiaoping Zheng, Yuan Cheng, and Ahmed Mebarki. 2019. A review of cellular automata models for crowd evacuation. *Physica A: Stat. Mech. Appl.* 526 (2019), 120752. https://doi.org/10.1016/j.physa.2019.03.117
[99] Dirk Helbing and Peter Molnar. 1995. Social force model for pedestrian dynamics. *Phys. Rev. E* 51, 5 (1995), 4282.
[100] Sepp Hochreiter and JÃ¼rgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.* 9, 8 (1997), 1735–1780.
[101] Jinghui Zhong, Yew Soon Ong, and Wentong Cai. 2016. Self-learning gene expression programming. *IEEE Trans. Evol. Comput.* 20, 1 (2016), 65–80.
[102] Forrest Stonedahl and Uri Wilensky. 2010. Finding forms of flocking: Evolutionary search in abm parameter-spaces. In *Proceedings of the International Workshop on Multi-agent Systems and Agent-based Simulation.* Springer, 61–75.
[103] Dahua Lin, Eric Grimson, and John Fisher. 2009. Learning visual flows: A lie algebraic approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 747–754.
[104] Dahua Lin, Eric Grimson, and John Fisher. 2010. Modeling and estimating persistent motion with geometric flows. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition.* 1–8.
[105] A. Lerner, Y. Chrysanthou, A. Shamir, and D. Cohen-Or. 2010. Context-dependent crowd evaluation. *Comput. Graph. Forum* 29, 7 (2010), 2197–2206. https://doi.org/10.1111/j.1467-8659.2010.01808.x
[106] L. Bruno, A. Corbetta, and A. Tosin. 2016. From individual behaviour to an evaluation of the collective evolution of crowds along footbridges. *J. Eng. Math.* 101, 1 (2016), 153–173.
[107] Rafael Pax and Jorge J. Gomez-Sanz. 2016. *A Greedy Algorithm for Reproducing Crowds.* Advances in Intelligent Systems and Computing, Vol. 473. 287–296. https://doi.org/10.1007/978-3-319-40159-1_24
[108] Claudio Feliciani and Katsuhiro Nishinari. 2016. An improved cellular automata model to simulate the behavior of high density crowd and validation by experimental data. *Physica A: Stat. Mech. Appl.* 451 (2016), 135–148. https://doi.org/10.1016/j.physa.2016.01.057
[109] D. Wolinski, S. J. Guy, A. H. Olivier, M. Lin, D. Manocha, and J. Pettre. 2014. Parameter estimation and comparative evaluation of crowd simulations. *Comput. Graph. Forum* 33, 2 (2014), 303–312. https://doi.org/10.1111/cgf.12328
[110] D. Wolinski, S. J. Guy, A. H. Olivier, M. C. Lin, D. Manocha, and J. Pettre. 2014. Optimization-based pedestrian model calibration for evaluation. In *Proceedings of the Conference on Pedestrian and Evacuation Dynamics*, Vol. 2. Elsevier Science Bv, 228–236.
[111] S. A. Stuvel, N. Magnenat-Thalmann, D. Thalmann, A. F. van der Stappen, and A. Egges. 2017. Torso crowds. *IEEE Trans. Visual. Comput. Graph.* 23, 7 (2017), 1823–1837.
[112] C. D. Boatright, M. Kapadia, J. M. Shapira, and N. I. Badler. 2015. Generating a multiplicity of policies for agent steering in crowd simulation. *Comput. Animat. Virt. Worlds* 26, 5 (2015), 483–494.
[113] A. Bera, S. Kim, and D. Manocha. 2016. Realtime anomaly detection using trajectory-level crowd behavior learning. *Proceedings of the 29th IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW'16)* (2016), 1289–1296. https://doi.org/10.1109/Cvprw.2016.163
[114] M. S. Zitouni, H. Bhaskar, J. Dias, and M. E. Al-Mualla. 2016. Advances and trends in visual crowd analysis: A systematic survey and evaluation of crowd modelling techniques. *Neurocomputing* 186 (2016), 139–159. https://doi.org/10.1016/j.neucom.2015.12.070
[115] S. Singh, M. Kapadia, P. Faloutsos, and G. Reinman. 2009. SteerBench: A benchmark suite for evaluating steering behaviors. *Comput. Anim. Virt. Worlds* 20, 5–6 (2009), 533–548. https://doi.org/10.1002/cav.277
[116] Vinicius J. Cassol, Estevao Smania Testa, Claudio Rosito Jung, Muhammad Usman, Petros Faloutsos, Glen Berseth, Mubbasir Kapadia, Norman I. Badler, and Soraia Raupp Musse. 2017. Evaluating and optimizing evacuation plans for crowd egress. *IEEE Comput. Graph. Appl.* 37, 4 (2017), 60–71. https://doi.org/10.1109/mcg.2017.3271454

[117] M. Zhao, W. Cai, and S. J. Turner. 2018. CLUST: Simulating realistic crowd behaviour by mining pattern from crowd videos. *Comput. Graph. Forum* 37, 1 (2018), 184–201. https://doi.org/10.1111/cgf.13259

[118] I. Karamouzas, N. Sohre, R. Hu, and S. J. Guy. 2018. Crowd space: A predictive crowd analysis technique. *ACM Trans. Graph.* 37, 6 (2018), 14. https://doi.org/10.1145/3272127.3275079

[119] B. Zhou, X. Tang, H. Zhang, and X. Wang. 2014. Measuring crowd collectiveness. *IEEE Trans. Pattern Anal. Mach. Intell.* 36, 8 (2014), 1586–1599.

[120] F. Martinez-Gil, M. Lozano, and F. Fernandez. 2017. Emergent behaviors and scalability for multi-agent reinforcement learning-based pedestrian models. *Simul. Model. Pract. Theory* 74 (2017), 117–133. https://doi.org/10.1016/j.simpat.2017.03.003

[121] Francisco Martinez-Gil, Miguel Lozano, and Fernando FernÃ¡ndez. 2014. MARL-Ped: A multi-agent reinforcement learning-based framework to simulate pedestrian groups. *Simul. Model. Pract. Theory* 47 (2014), 259–275. https://doi.org/10.1016/j.simpat.2014.06.005

[122] Francisco Martinez-Gil, Miguel Lozano, and Fernando FernÃ¡ndez. 2012. Calibrating a motion model based on reinforcement learning for pedestrian simulation. In *Motion in Games*, Marcelo Kallmann and Kostas Bekris (Eds.). Springer, Berlin, 302–313.

[123] Nan Hu, Jinghui Zhong, Joey Tianyi Zhou, Suiping Zhou, Wentong Cai, and Christopher Monterola. 2018. Guide them through: An automatic crowd control framework using multi-objective genetic programming. *Appl. Soft Comput.* 66 (2018), 90–103. https://doi.org/10.1016/j.asoc.2018.01.037

[124] Yanyu Xu, Zhixin Piao, and Shenghua Gao. 2018. Encoding crowd interaction with deep neural network for pedestrian trajectory prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 5275–5284.

[125] Hao Xue, Du Q. Huynh, and Mark Reynolds. 2018. SS-LSTM: A hierarchical LSTM model for pedestrian trajectory prediction. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV'18).* 1186–1194.

[126] Raymond L. Major. 2000. A complexity model and a polynomial algorithm for decision tree-based feature construction. *Comput. Intell.* 16, 1 (2000), 53–78.

[127] K. Neshatian, M. Zhang, and P. Andreae. 2012. A filter approach to multiple feature construction for symbolic learning classifiers using genetic programming. *IEEE Trans. Evolution. Comput.* 16, 5 (2012), 645–661.

[128] J. Thomas and L. Sael. 2015. Overview of integrative analysis methods for heterogeneous data. In *Proceedings of the International Conference on Big Data and Smart Computing (BIGCOMP'15).* 266–270. https://doi.org/10.1109/35021BIGCOMP.2015.7072811

[129] Lei Yuan, Yalin Wang, Paul M. Thompson, Vaibhav A. Narayan, Jieping Ye, Alzheimer's Disease Neuroimaging Initiative et al. 2012. Multi-source feature learning for joint analysis of incomplete multiple heterogeneous neuroimaging data. *NeuroImage* 61, 3 (2012), 622–632.

[130] Q. Shi, C. Zhang, M. Peng, X. Yu, T. Zeng, J. Liu, and L. Chen. 2017. Pattern fusion analysis by adaptive alignment of multiple heterogeneous omics data. *Bioinformatics* 33, 17 (2017), 2706–2714.

[131] Jaya Thomas and Lee Sael. 2016. Maximizing information through multiple kernel-based heterogeneous data integration and applications to ovarian cancer. In *Proceedings of the 6th International Conference on Emerging Databases Technologies, Applications, and Theory.* 95–98. https://doi.org/10.1145/3007818.3007831

[132] Srivastav R. Janapalli, Omar Hesham, and Gabriel A. Wainer. 2019. Heterogeneous crowd simulation. In *Proceedings of the Annual Simulation Symposium.*

[133] Liping Zheng, Qin Dang, Yajun Cheng, Wang Lin, and Li Lin. 2012. Simulating heterogeneous crowds from a physiological perspective. *Neurocomputing* 172, C (2012), 180–188.

[134] Stephen J. Guy, Sujeong Kim, Ming C. Lin, and Dinesh Manocha. 2011. Simulating heterogeneous crowd behaviors using personality trait theory. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation.* 43–52.

[135] Been Kim Finale Doshi Velez. 2017. Towards a rigorous science of interpretable machine learning. Retrieved from https//arXiv.org/abs/1702.08608v2.

[136] Sunrise Wang, James Edward Gain, and Geoff Stuart Nistchke. 2015. Controlling crowd simulations using neuro-evolution. In *Proceedings of the Annual Conference on Genetic and Evolutionary Computation (GECCO'15).* ACM, New York, NY, 353–360. https://doi.org/10.1145/2739480.2754715

[137] N. Lee, W. Choi, P. Vernaza, C. B. Choy, P. H. S. Torr, and M. Chandraker. 2017. DESIRE: Distant future prediction in dynamic scenes with interacting agents. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'17).* 2165–2174.

[138] Shuai Yi, Hongsheng Li, and Xiaogang Wang. 2016. Pedestrian behavior understanding and prediction with deep neural networks. In *Proceedings of the European Conference on Computer Vision (ECCV'16)*, Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling (Eds.). Springer, Cham, 263–279.

[139] Hang Su, Jun Zhu, Yinpeng Dong, and Bo Zhang. 2017. Forecast the plausible paths in crowd scenes.. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'17)*. 2772–2778.

[140] Suna Yin, Dehua Chen, and Jiajin Le. 2017. Deep neural network based on translation model for diabetes knowledge graph. In *Proceedings of the 5th International Conference on Advanced Cloud and Big Data (CBD'17)*. 318–323.

[141] R. Krishnan, G. Sivakumar, and P. Bhattacharya. 1999. Extracting decision trees from trained neural networks. *Pattern Recogn.* 32, 12 (1999).

[142] Ulf Johansson and Lars Niklasson. 2009. Evolving decision trees using oracle guides. In *Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining*. 238–244.

[143] Olcay Boz. 2002. Extracting decision trees from trained neural networks. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'02)*. Association for Computing Machinery, New York, NY, 456–461. https://doi.org/10.1145/775047.775113

[144] A. Lensen, B. Xue, and M. Zhang. 2020. Genetic programming for evolving a front of interpretable models for data visualization. *IEEE Trans. Cybernet.* (2020), 1–15. https://ieeexplore.ieee.org/abstract/document/9007046.

[145] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. 2018. A survey of methods for explaining black box models. *ACM Comput. Surv.* 51, 5, Article 93 (Aug. 2018), 42 pages.

[146] Zhanjing Wang, Hongze Yin, and Fachao Li. 2011. The computing method for core attributes based on decision tree. In *Proceedings of the International Conference on Machine Learning and Cybernetics*, Vol. 1. 119–124.

[147] Salvador Garcıa, Alberto Fernández, and Francisco Herrera. 2009. Enhancing the effectiveness and interpretability of decision tree and rule induction classifiers with evolutionary training set selection over imbalanced problems. *Appl. Soft Comput.* 9, 4 (2009), 1304–1314.

[148] Leonardo Vanneschi, Mauro Castelli, and Sara Silva. 2010. Measuring bloat, overfitting and functional complexity in genetic programming. In *Proceedings of the Genetic and Evolutionary Computation Conference*.

[149] Nam Le, Hoai Nguyen Xuan, Anthony Brabazon, and Thuong Pham Thi. 2016. Complexity measures in genetic programming learning: A brief review. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC'16)*. 2409–2416.

[150] Andrei Lissovoi and Pietro S. Oliveto. 2018. On the time and space complexity of genetic programming for evolving boolean conjunctions. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*.

[151] Ji Ni and Peter Rockett. 2014. Tikhonov regularization as a complexity measure in multiobjective genetic programming. *IEEE Trans. Evolution. Comput.* 19, 2 (2014), 157–166.

[152] Bolei Zhou, Xiaogang Wang, and Xiaoou Tang. 2012. Understanding collective crowd behaviors:Learning a mixture model of dynamic pedestrian-agents. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'12)*.

[153] Golnaz Habibi, Nikita Jaipuria, and Jonathan P. How. 2019. Incremental learning of motion primitives for pedestrian trajectory prediction at intersections. arXiv: Robotics.

[154] Govea and Alejandro Dizan Vasquez. 2010. Incremental learning for motion prediction of pedestrians and vehicles, vol. 64. Springer tracts in advanced robotics, 10.1007/978-3-642-13642-9.

[155] Anoop Aroor, Susan L. Epstein, and Raj Korpan. 2018. Online learning for crowd-sensitive path planning. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS'18)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 1702–1710.

[156] Omar Hesham and Gabriel Wainer. 2016. Centroidal particles for interactive crowd simulation. In *Proceedings of the Summer Computer Simulation Conference (SCSC'16)*. Society for Computer Simulation International, San Diego, CA, Article 7, 8 pages.

[157] Nagaraj Joshi. 2018. Crowd simulation with panic mode. [EB/OL]. Retrieved from https://assetstore.unity.com/packages/templates/packs/crowd-simulation-with-panic-mode-122552.

[158] Aligned Games. 2019. Crowd Simulation System. [EB/OL]. Retrieved from https://assetstore.unity.com/packages/templates/systems/crowd-simulation-system-143299.

[159] Seagull Entertainment. 2019. NaviPro Pathfinding. [EB/OL]. Retrieved from https://assetstore.unity.com/packages/tools/ai/navipro-pathfinding-155501.

[160] A. Best, S. Narang, and D. Manocha. 2016. Real-time reciprocal collision avoidance with elliptical agents. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'16)*. 298–305. https://doi.org/10.1109/ICRA.2016.7487148

[161] D. Vasquez, B. Okal, and K. O. Arras. 2014. Inverse reinforcement learning algorithms and features for robot navigation in crowds: An experimental comparison. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. 1341–1346. https://doi.org/10.1109/IROS.2014.6942731

[162] Weichen Liao, Armel U. Kemloh Wagoum, and Nikolai W. F. Bode. 2017. Route choice in pedestrians: Determinants for initial choices and revising decisions. *J. Roy. Soc. Interface* 14, 127 (2017), 20160684.

[163] Liu Wenlong, Zhang Jing, Zhou Suiping, and Li Yuelong. 2018. Simulation and optimization of crowd movement behavior based on repast simphony platform. *Chinese J. Comput. Sci.* 45, 12 (2018), 187–191.

[164] Jan Procházka, Richard Cimler, and Kamila Olševičová. 2015. Pedestrian modelling in netlogo. In *Emergent Trends in Robotics and Intelligent Systems*, Peter Sinčák, Pitoyo Hartono, Mária Virčíková, Ján Vaščák, and Rudolf Jakša (Eds.). Springer, Cham, 303–312.

[165] Colin M. Henein and Tony White. 2005. Agent-based modelling of forces in crowds. In *Multi-agent and Multi-agent-based Simulation*, Paul Davidsson, Brian Logan, and Keiki Takadama (Eds.). Springer, Berlin, 173–184.

[166] Neal Wagner and Vikas Agrawal. 2014. An agent-based simulation system for concert venue crowd evacuation modeling in the presence of a fire disaster. *Expert Syst. Appl.* 41, 6 (2014), 2807–2815. https://doi.org/10.1016/j.eswa.2013.10.013

[167] Mehdi Moussaïd, Dirk Helbing, and Guy Theraulaz. 2011. How simple rules determine pedestrian behavior and crowd disasters. *Proc. Natl. Acad. Sci. U.S.A.* 108, 17 (2011), 6884–6888.

[168] Sean Curtis, Andrew Best, and Dinesh Manocha. 2016. Menge: A modular framework for simulating crowd movement. *Collect. Dynam.* 1, 0 (2016), 1–40. https://doi.org/10.17815/CD.2016.1

[169] Benedikt Kleinmeier, Benedikt Zönnchen, Marion Gödel, and Gerta Köster. 2019. Vadere: An open-source simulation framework to promote interdisciplinary understanding. Retrieved from https//arXiv.org/abs/1907.09520.

[170] Christian Gloor. 2016. PedSim: Pedestrian crowd simulation. Retrieved from http://pedsim.silmaril.org.

[171] Mohcine Chraibi and Jun Zhang. 2016. JuPedSim: An open framework for simulating and analyzing the dynamics of pedestrians. In *Proceedings of the Traffic, Mobility, and Logistics Conference (SUMO'16)*. Berichte aus dem DLR-Institut für Verkehrssystemtechnik, Vol. 30, 127–134.

[172] Vaisagh. 2014. Java-based motion planning testbed. https://github.com/vaisaghvt/MoPlaT.

[173] Michael J. North, Nicholson T. Collier, Jonathan Ozik, Eric R. Tatara, Charles M. Macal, Mark Bragen, and Pam Sydelko. 2013. Complex adaptive systems modeling with repast simphony. *Complex Adapt. Syst. Model.* 1, 1 (Mar. 2013), 3. https://doi.org/10.1186/2194-3206-1-3

[174] Uri Wilensky. 2014. NetLogo: Center for connected learning and computer-based modeling, Northwestern University, Evanston, IL.

Received October 2019; revised July 2021; accepted August 2021